

Санкт-Петербургский Государственный Университет
Экономический факультет
Кафедра информационных систем в экономике

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по направлению – «Бизнес-информатика»

РАЗРАБОТКА ПРОТОТИПА БАЗЫ ДАННЫХ ДЛЯ СБОРА
ИНФОРМАЦИИ ОБ ЭФФЕКТИВНОСТИ РАБОТЫ СОТРУДНИКОВ
И ЕЁ АНАЛИЗА С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ НЕЧЕТКОГО
ВЫВОДА

Выполнил:

Бакалавриант 4 курса, группы БИ-4

Вяткина Мария Валерьевна

_____/Подпись/

Научный руководитель:

доцент кафедры информационных систем в
экономике экономического факультета
СПбГУ, Забоев Михаил Валерьевич

_____/Подпись/

Санкт-Петербург

2018

Оглавление

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ОБЩИЕ ПОДХОДЫ К ПРОЕКТИРОВАНИЮ БАЗ ДАННЫХ В MS ACCESS	6
1.1. Разновидности моделей данных	7
1.2. Проектирование реляционных баз данных.....	10
ГЛАВА 2. МЕТОДЫ АНАЛИЗА ДАННЫХ ДЛЯ ОЦЕНКИ ПОКАЗАТЕЛЕЙ ЭФФЕКТИВНОСТИ РАБОТЫ СОТРУДНИКОВ ОРГАНИЗАЦИИ	14
2.1. Методы кластерного анализа	14
2.2. Основы теории нечетких множеств.....	21
2.3. Использование искусственных нейронных сетей	26
ГЛАВА 3. ПРОЕКТИРОВАНИЕ ПРОТОТИПА БАЗЫ ДАННЫХ В MS ACCESS	31
3.1. Описание ограничений предметной области.....	31
3.2. Определение требований к системе через информационные потребности пользователей	32
3.3. Реализация проекта в среде СУБД MS ACCESS, включая создание пользовательского приложения	34
ГЛАВА 4. ПОСТРОЕНИЕ СИСТЕМЫ НЕЧЕТКОГО ВЫВОДА СРЕДСТВАМИ MATLAB	36
4.1. Связь базы и системы нечеткого вывода	36
4.2. Определение лингвистических переменных для системы нечеткого вывода и создание блоков переменных	39
4.3. Описание работы экспертной системы в пакете MATLAB.	40
ЗАКЛЮЧЕНИЕ	43
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	44
ПРИЛОЖЕНИЯ.....	47

ВВЕДЕНИЕ

В настоящее время ни для кого не секрет, что цель любой компании как производителя товаров или услуг – максимизация прибыли. При этом прибыль напрямую зависит от уровня цен, величины издержек, объема продаж. И конечно она зависит от той команды, которая работает над продуктом – от сотрудников.

При этом, для того, чтобы корректно оценивать вклад каждого сотрудника в отдельности необходимо периодически проводить анализ результатов их работы и оценку индивидуальной эффективности.

KPI (Key Performance Indicators) Ключевые показатели эффективности – это система оценки, которая помогает организации определить достижение стратегических и тактических целей. Их использование дает организации возможность оценить свое состояние и помочь в оценке реализации стратегий. KPI позволяет производить контроль деловой активности сотрудников и компании в целом в реальном времени [21].

Целесообразно собирать информацию и проводить оценку эффективности сотрудников, основываясь на результатах достижения ими личных KPI по разным параметрам. Однако на данный момент во многих компаниях оценка производится преимущественно в формате достижения или не достижения поставленной цели, без учета того, на сколько именно сотрудник приблизился к достижению цели, и не учитывает слабо формализуемых факторов (таких как сложность работ, загруженность работника и т.д.), которые всегда присутствуют. Сбор информации для анализа зачастую требует множества одинаковых действий, которые можно оптимизировать. А информация хранится в разнородном виде: каждый ведет требуемые записи, но разрозненно и без соблюдения стандартов, меняя формат и структуру. В связи с этим логика изложения оказывается противоречивой, что усложняет проведение анализа.

В то же время, единство шаблона для ввода и хранения данных можно обеспечить единой базой данных, а для последующей работы с собранной информацией использовать средства интеллектуального анализа данных. В частности, нечетко-множественный анализ способен дать оценку степени достижения целей, а также учесть плохо формализуемые критерии, путем введения экспертных оценок.

Таким образом, актуальность работы обосновывается:

- Отсутствием во многих организациях единого инструмента для сбора данных о целях и задачах сотрудников на предстоящий финансовый год, а также данных о фактических результатах, достигнутых по окончании каждого месяца/квартала;

- Отсутствием прозрачности в каскадировании задач руководителями (группы, отдела, организации) с одной стороны и принятии их подчиненными с другой.
- А также отсутствием системы оценки достижения КРІ сотрудниками, учитывающей все критерии в многофакторном анализе.

Стоит также отметить, что эффективность работы сотрудников необходимо оценивать и для правильного распределения обязанностей; корректного делегирования более эффективным сотрудникам более сложных задач; для справедливого повышения по карьерной лестнице в связи со стабильным перевыполнением КРІ, либо внедрения мер по повышению эффективности при низком показателе. Применение методов нечеткой логики в совокупности с единым инструментом хранения данных позволяет хранить полную информацию о ежемесячных/квартальных результатах работы сотрудников и сравнивать эти результаты с плановыми, проводя более точную оценку работы подчиненных (в сравнении с принятой на текущий момент).

Объектом исследования моей практической работы являются сотрудники организации, предметом - анализ эффективности работы сотрудников организации.

Цель - создать прототип базы данных для централизованного хранения показателей эффективности сотрудников и разработать систему нечеткого вывода итоговых оценок их работы. Для достижения данной цели поставлены следующие задачи:

1. Разработать прототип БД с возможностью каскадирования поставленных задач в цели подчиненных сотрудников;
2. Сформулировать лингвистические переменные на основании которых будет проводиться анализ, сформировать блоки с базами правил;
3. Разработать всю систему нечеткого вывода;
4. Протестировать работу системы, а также одного выбранного блока в отдельности на разных значениях переменных;
5. Представить результат анализа в виде отчета для менеджеров.

Средой разработки базы данных был выбран MS Access (преимущественно использовались макросы Visual Basic и запросы SQL), так как он в большей степени подходит для работы с базами данных: для формирования таблиц и установления связей между ними, для удобного внесения данных с возможностью их выведения в других удобных форматах для визуализации. Использование средств Visual Basic позволяет гибко настраивать работу системы. Логическая структура данных, иначе говоря ИЛМ – информационно-логическая модель, в выбранной программе проста и удобна в понимании и реализации разработчику, а также во внесении корректировок при необходимости. Базы

данных Access на данном этапе вполне достаточно даже для компании высокого уровня, так как в пилотном варианте ее использование планируется только одним определенным отделом, соответственно количество «юзеров» (лиц, работающих с базой) не будет превышать 50 сотрудников, таким образом, база не будет сильно перегружаться от объема хранимых данных. Если же возникнет необходимость масштабирования, то при некоторых поправках в структуре таблиц и схемы данных, реализация быстро работающего продукта тоже будет возможна.

Средой для анализа показателей эффективности был выбран графический интерфейс FIS в MATLAB, а также система нечеткого вывода в MATLAB Simulink. В терминах задания нечетких чисел, отвечающих выбранным лингвистическим переменным, составления блоков лингвистических переменных, а также итогового связывания этих блоков в единую систему нечеткого вывода MATLAB обладает удобным инструментарием и простой и понятной схематичной визуализацией структуры нечеткого вывода и результатов вывода. В то же время предоставляется возможность программной реализации, как альтернатива графическому интерфейсу (вариант реализации – выбор аналитика). Помимо указанных преимуществ обладает также интуитивно понятным пользовательским интерфейсом. Среди возможных альтернатив рассматривались: Python, C++ (Fuzzy Logic Control library, Fuzzy Lite), FuzzyTech, Mathcad.

ГЛАВА 1. ОБЩИЕ ПОДХОДЫ К ПРОЕКТИРОВАНИЮ БАЗ ДАННЫХ В MS ACCESS

Первым шагом на пути к появлению баз данных было появление жестких дисков – винчестеров в конце 60-х. 70-е годы ознаменовались подготовкой теоретической базы к созданию будущих хранилищ данных. В начале 80-х годов уже начали появляться производительные программные средства для построения информационных систем (ИС). Столь быстрое развитие информационных технологий привело к тому, что к хранению, обработке и представлению данных стали предъявлять все больше требований. Процесс развития ИТ, а значит и требований к работе с данными, не закончен и на текущий момент времени, что приводит к необходимости изучения и развития теории баз данных.

Итак, базы данных создаются специально для хранения, обработки, проведения расчётов, сортировки, выборки и представления любых массивов данных по любым критериям. При этом сами базы могут различаться по модели данных:

- Иерархические;
- Сетевые;
- Реляционные;
- Смешанные;
- Мультимодельные.

Отдельно выделяют сверхбольшие базы данных, объем которых в 1997 году был равен 7 терабайт (Knight Ridder's DIALOG [35]), в 2006 - 850 терабайт (БД поисковой машины Google [28]), а к 2014 году оценивался в 10—15 эксабайт (так же БД поисковой машины Google [29]).

1.1. Разновидности моделей данных

Классификация моделей данных была представлена выше, опишем более подробно некоторые из них.

Иерархическая модель

Объекты, связанные иерархически, образуют ориентированный граф (Рис. 1).

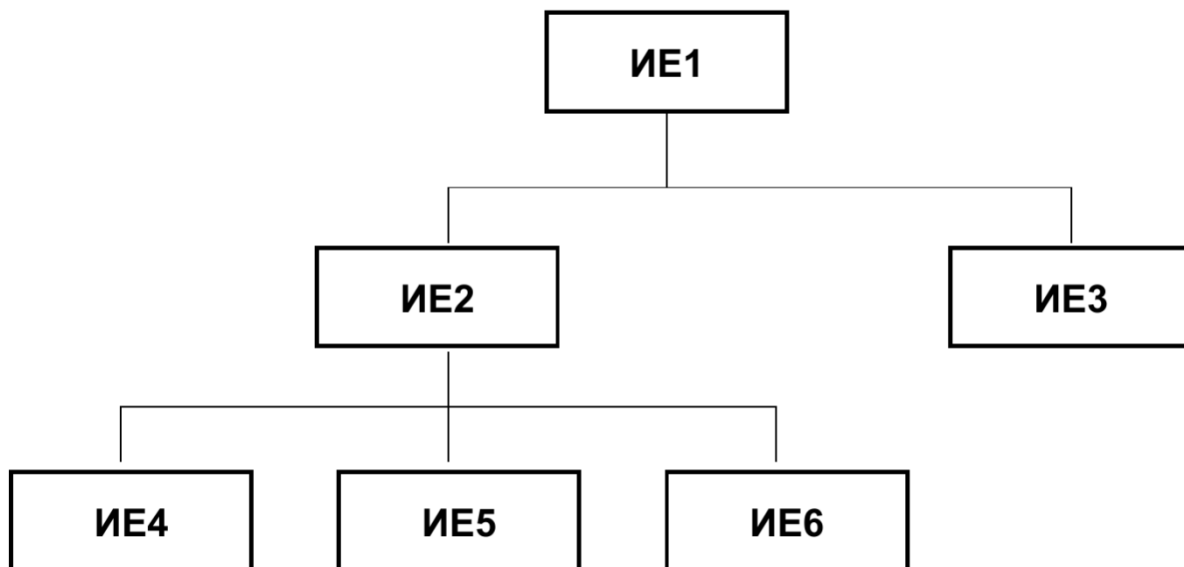


Рис. 1. Иерархическая модель данных. ИЕ- иерархическая единица.

Основными понятиями иерархической структуры являются: узел, уровень и связь. Узел – совокупность атрибутов, описывающих некоторый объект. При этом каждый потомок (узел, находящийся на более низком уровне) имеет только одного предка (связан только с одним узлом, находящимся на более высоком уровне). Основные манипуляции при этом:

- Поиск элемента в базе данных;
- Переход от одного дерева к другому;
- Перемещение от записи к записи внутри дерева;
- Вставка записи;
- Удаление элемента.

Среди достоинств такой организации данных эффективное использование памяти и сравнительно высокая скорость выполнения операций. Из недостатков громоздкость и сложность физической реализации для больших древовидных структур.

Сетевая модель

Рассматривается сетевая модель с однотипными файлами (Рис. 2).

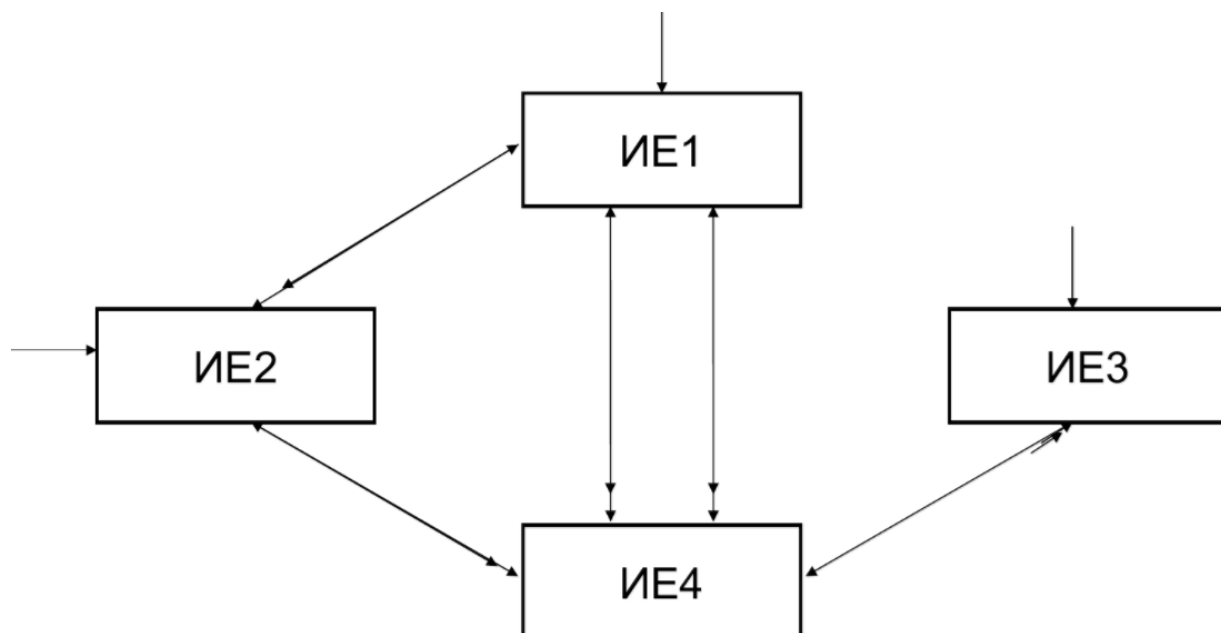


Рис. 2. Сетевая модель. ИЕ- иерархическая единица.

В такой этой модели данные представлены как коллекция записей, связи формируют наборы. Сетевая модель – это граф с записями в виде узлов графа и наборами в виде его ребер. В основу положены графы произвольной структуры, отражающие взаимосвязи между данными.

В отличие от иерархической модели в данном случае на формирование связей не накладываются ограничения (то есть каждый потомок может иметь больше одного предка и наоборот).

В данном случае основными операциями будут:

- Поиск в базе данных некоторого элемента;
- Переход от предка к потомку;
- Переход от потомка к предку;
- Вставка записи;
- Удаление записи и др.

Достоинства такой модели: при работе с данными память используется эффективно; широк перечень решаемых задач, благодаря наличию разных связей. Недостатки следующие:

физическая реализация сложна; работать с данными не удобно, так как связи между элементами «жесткие»; контроль целостности связей слаб.

Реляционная модель

Реляционная модель представима следующим образом (Рис. 3):

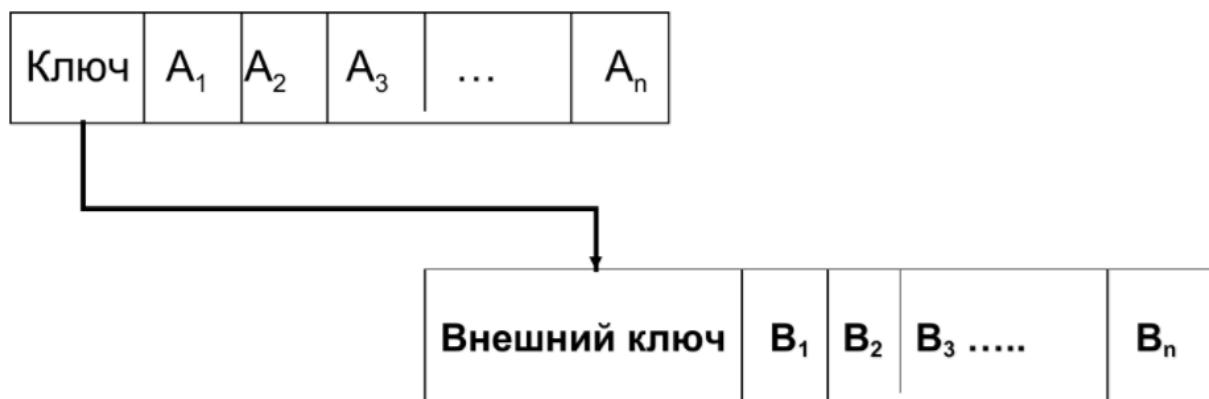


Рис. 3. Реляционная модель

Реляционной базой данных называется база, построенная на реляционной модели [30, 31]. Реляционной моделью, в свою очередь, называется совокупность двумерных таблиц данных. В реляционной базе каждому объекту соответствует строка в таблице. Таблицы базы могут быть связаны посредством схемы данных или же «на неформальном уровне», где связь только подразумевается, но не отражается структурно. Представляется реляционная база данных как совокупность строк и столбцов, где первые соответствуют экземпляру объекта, а вторые – атрибутам (признаки, характеристики).

Классическая реляционная модель подразумевает неделимость данных, которые хранятся в полях таблиц. Для этого существует процедура нормализации – представление информации в таблице в первой нормальной форме.

В целом, реляционные базы данных характеризуются возможностью оперативного доступа к хранящимся данным, надежностью и целостностью данных, отсутствием избыточности.

Среди достоинств модели простота моделирования и физической реализации, наглядность представления информации, а также эффективность ее обработки. Недостаток заключается в том, что обеспечить непротиворечивость и целостность данных, которые хранятся в базе достаточно трудно.

Для хранения информации о деятельности предприятий разного размера (малые, средние, крупные) как правило не требуется сверхбольших баз. В таких случаях наиболее часто используются возможности Microsoft Access. При этом создаются базы самой различной тематики: база для хранения данных органов опеки и попечительства, описанная в статье [16], база данных Шанучского месторождения (Камчатка) [13], автозаправки [9], гидрогеологических параметров артезианских скважин [12] и т.д.

В данной работе представлена реляционная база данных, не относящаяся к группе сверхбольших, предназначенная для хранения целей и стратегий сотрудников на предстоящий год, а также фактически достигнутых результатов.

1.2. Проектирование реляционных баз данных

Перед тем, как начать проектировать базу данных, необходимо определиться с ее структурным наполнением, описать ограничения предметной области. Важно также определить пользователей, для которых эта база разрабатывается, и определить их информационные потребности; сформулировать требования к базе (в результате проектирования требования, а значит и информационные потребности должны быть удовлетворены). На этом же этапе описывается схема документооборота.

На начальных этапах разработки проводится анализ входной и выходной информации, определение содержимого справочников будущей базы, содержащих ключевые поля, по которым таблицы будут связаны. При необходимости, задаются так же формы оперативных документов.

Далее происходит моделирование данных: выделяются информационные объекты (сущности) и их атрибуты. Сущность – это реальный либо представляемый объект, информация о котором должна сохраняться и быть доступна. При этом, каждая сущность является множеством объектов, называемых экземплярами сущности. Атрибут – это определенное, далее неделимое свойство сущности. Совокупность значений атрибутов определяет экземпляр сущности. Также необходимо пояснить понятия ключа и связи: ключ – атрибут, либо группа атрибутов, однозначно идентифицирующая экземпляр сущности; связь – логическое соотношение между двумя сущностями. В базе данных сущность – таблица, экземпляр сущности – строка в таблице, атрибут – колонка таблицы.

На текущий момент моделирования данные в информационных объектах могут дублироваться, а значит, занимать неоправданно больше места в хранилище. Для того чтобы

избежать дублирования, проводится нормализация информационных объектов, основывающаяся на трех правилах:

1. Приведение к первой нормальной форме (1НФ). Сущность находится в 1НФ тогда, когда все атрибуты содержат атомарные значения. То есть среди атрибутов не найдется нескольких значений для одного экземпляра;
2. Приведение ко второй нормальной форме (2НФ). Сущность находится во 2НФ, если она в 1НФ и каждый неключевой реквизит функционально зависит от ключа, а зависимости от части ключа нет. Для приведения сущности к 2НФ необходимо выделить атрибуты, зависящие от части ключа и поместить их в новую сущность;
3. Приведение к третьей нормальной форме (3НФ). Сущность находится в 3НФ, если она в 2НФ и никакой неключевой атрибут не зависит от другого неключевого.

Необходимо пояснить, что означает «функциональная зависимость»: реквизит В функционально зависит от реквизита А тогда и только тогда, когда конкретному значению А соответствует одно единственное значение В, то есть А однозначно определяет В. На схеме это выглядит следующим образом (Рис. 4):

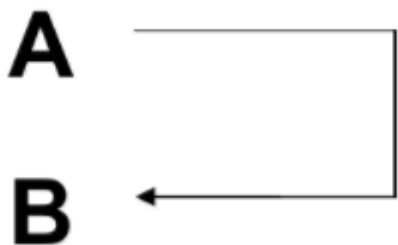


Рис. 4. Реквизит А определяет реквизит В

Рассмотрим пример. Пусть, в некоторой базе данных предполагается хранить информацию о сотрудниках. Первоначально созданная сущность содержит необходимую информацию, но правило 2НФ нарушается (Рис. 5).



Рис. 5. Не соблюдается требование 2НФ

В данном случае необходимо выделить информацию о должностях в отдельную таблицу для корректной нормализации.

После нормализации выделенных объектов можно приступить к составлению информационно-логической модели (ИЛМ). ИЛМ – это прообраз будущей схемы данных в самой базе, показывающий связь выделенных объектов (таблиц). Например, таблица с информацией о сотрудниках и таблица с информацией о должностях сотрудников будут связаны следующим образом (Рис. 6):



Рис. 6. Связь двух объектов

В данном случае выбрана связь «один-ко-многим», поскольку на одну должность могут быть наняты несколько сотрудников (2 программиста, 3 оператора). Всего существует три типа связей:

1. Один-к-одному;
2. Многие-ко-многим;
3. Один-ко-многим.

Вся ИЛМ выглядит, как несколько объектов (все, которые были выделены), соединенных связями. Например, ИЛМ базы данных спортивной школы может выглядеть так (Рис. 7):

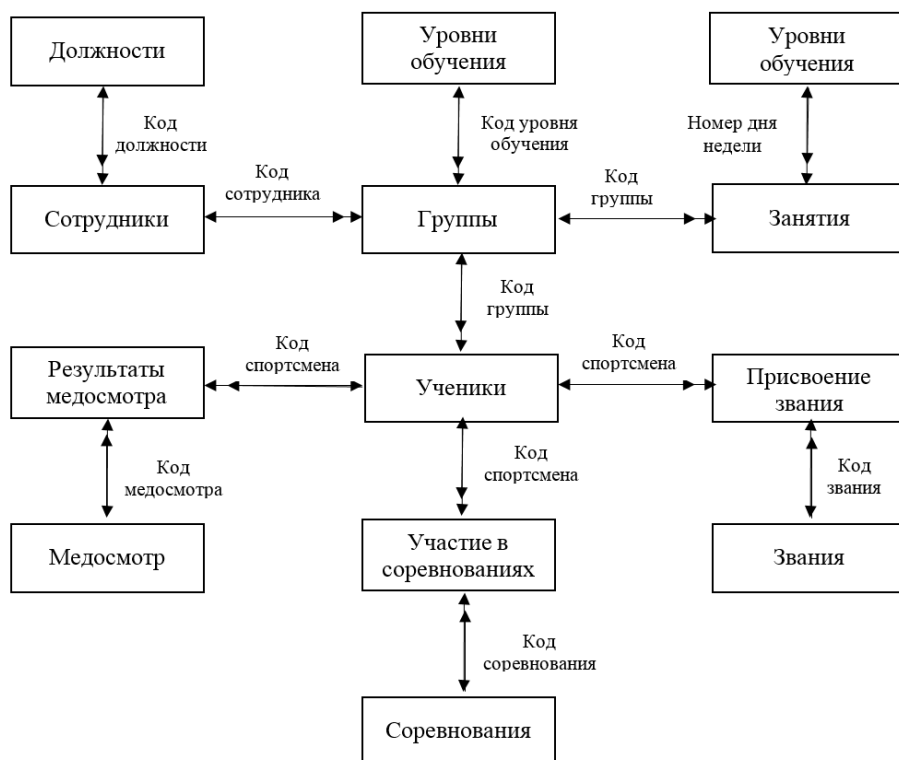


Рис. 7. Пример ИЛМ

Здесь каждый объект – отдельная таблица с набором атрибутов.

Последним шагом перед тем, как приступить к реализации проекта в прикладной программе, является определение типов переменных. Чтобы предотвратить внесение пользователем некорректных данных, разработчику необходимо определить атрибутам сущностей соответствующие типы: например, атрибуту «Дата» указать тип «Дата/Время», атрибуту «Телефон» - «Числовой» и т.д.

Далее можно приступить к реализации в прикладном пакете. В данной работе база данных реализована в MS Access.

ГЛАВА 2. МЕТОДЫ АНАЛИЗА ДАННЫХ ДЛЯ ОЦЕНКИ ПОКАЗАТЕЛЕЙ ЭФФЕКТИВНОСТИ РАБОТЫ СОТРУДНИКОВ ОРГАНИЗАЦИИ

В данной главе будут предложены методы анализа данных для оценки показателей эффективности работы сотрудников на основе искусственных нейронных сетей и теории нечетких множеств, которые, с одной стороны, позволят быстро провести самостоятельный анализ, что, в свою очередь, позволит существенно сократить временные, а значит и финансовые затраты по сравнению с использованием работы персонала, ответственного за оценку KPI и продвижение по карьерной лестнице, а с другой стороны, предоставят более полную и точную базу критериев и правил оценки работы сотрудников, по сравнению с методами, которые не учитывают имеющуюся информацию о плохо формализуемых факторах.

В рамках предложенных в данной работе методов оценки показателей эффективности работы сотрудников будут рассматриваться такие задачи интеллектуального анализа данных, как кластеризация, прогнозирование, а также построение системы нечеткого вывода. Для решения данных задач будут применяться методы с использованием нейронных сетей и теории нечетких множеств.

2.1. Методы кластерного анализа

Кластерным анализом называется совокупность методов и алгоритмов, которые предназначены для разбиения исходной совокупности рассматриваемых объектов на группы таким образом, чтобы в каждую отдельную группу (кластер) попадали объекты с близкими значениями свойств и признаков [5]. Задача кластеризации является одной из основных задач в области анализа данных.

Кластеризация используется во многих областях: здравоохранение, страхование, маркетинг, борьба с мошенничеством, прогнозирование, банковская деятельность и многие другие. Весь список областей применения кластерного анализа достаточно широк, приведем некоторые примеры:

- здравоохранение – кластеризация заболеваний, их симптомов, способов лечения, классификация групп пациентов;

- страхование – разбиение клиентов на кластеры, каждый из которых соответствует определенной рисковой группе;
- маркетинг – сегментация рынка по группам товаров или потребителей, определение потенциального потребителя;
- банковская деятельность – кластеризация клиентов банка при постановке вопроса о выдаче кредита;
- работа с персоналом – деление сотрудников на группы на основании результатов проделанной работы с целью выявления претендентов на повышение либо понижение;

Приведенные примеры применения кластерного анализа данных находят отражение в статьях российских и зарубежных авторов, освещающих различную проблематику выбранной области. В статье [10] рассматриваемый метод интеллектуального анализа данных используют для оценки перспектив вхождения российских университетов в первую сотню ведущих мировых университетов. В статье [19] используются методы кластеризации при оценке эффективности стратегического управления предприятиями региона, делая акцент на иерархический метод. Согласно позиции автора, в однородных предприятиях региона, объединенных в типы, действуют близкие по содержанию закономерности развития.

Существует две группы методов кластерного анализа: иерархические и неиерархические. Основные этапы кластерного анализа для обеих групп одинаковы:

1. Формирование базы данных (в частности определение характеристик рассматриваемых объектов: количественных, качественных). На данном этапе происходит предварительная обработка данных: форматирование, структурирование, нормализация, желательно также сокращение размерности данных, если это возможно.
2. Выбор метрик (определение критериев схожести объектов): определение степени схожести объектов (по совокупности их характеристик) в количественной форме. Наиболее часто используется Евклидова метрика в силу простоты её понимания и представления;
3. Разбиение всей совокупности объектов на группы;
4. Интерпретация результатов;
5. Оценка результатов.

Иерархические методы

В иерархических методах кластеризации происходит построение полного дерева вложенных кластеров, что избавляет от необходимости определять количество кластеров заранее. На первом шаге каждый объект принимается за отдельный кластер. Далее на каждом шаге похожие по признакам объекты объединяются в новые кластеры. Иерархические методы кластеризации различаются правилами построения кластеров. В качестве правил выступают критерии, которые используются для вычисления расстояния между объектами.

Недостатками иерархических методов кластеризации является отсутствие возможности работать с большим объемом данных. Преимуществом является отсутствие необходимости заранее определять количество кластеров, а также получение в результате кластеризации полного дерева вложенных кластеров, наглядно отражающего структуру данных.

В качестве примера визуализации результата можно привести дендрограмму, которая помимо визуализации позволяет определить необходимое количество кластеров. Пример дендрограммы, где задана степень сходства объектов, указан ниже (Рис. 8):

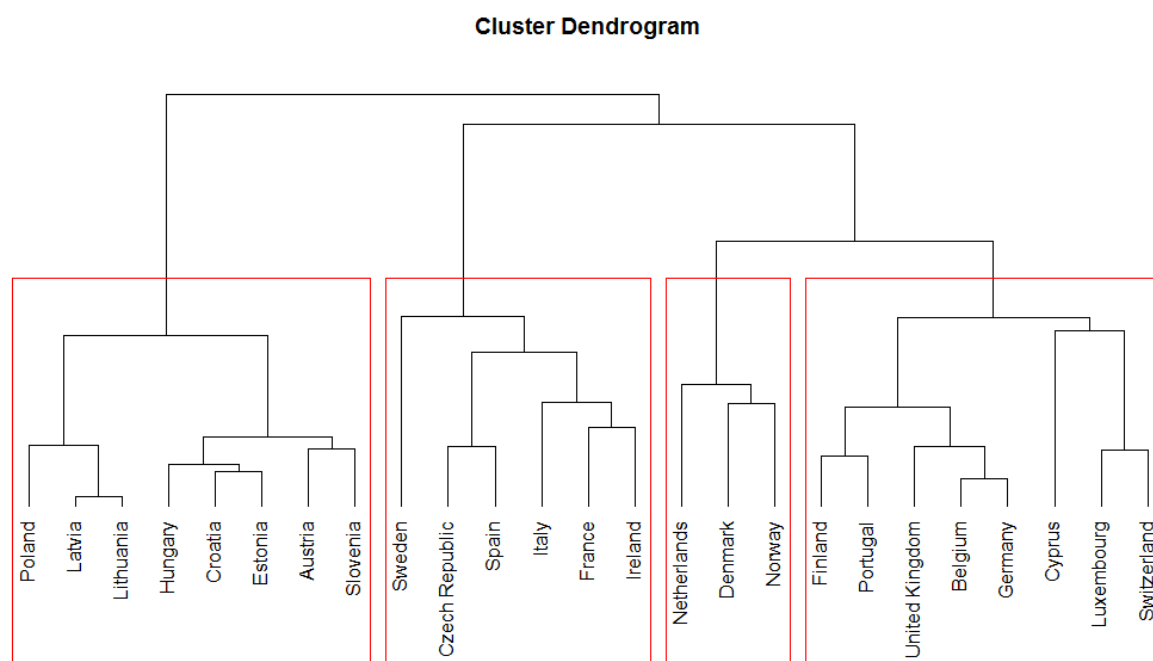


Рис. 8. Дендрограмма. Пример кластеризации стран Европы.

Ветвление начинается с 2 кластеров, далее делится на 3, 4 и так далее. Красным цветом выделено деление на 4 кластера.

Неиерархические методы

Неиерархические методы обладают большей устойчивостью к шумам и выбросам, некорректному выбору метрики, включению незначимых переменных в набор, участвующий в кластеризации [25]. Однако так как в данном случае результатом является единственное разделение на кластеры, то при использовании неиерархических методов необходимо заранее определить количество кластеров.

Самый распространенный из методов – алгоритм k-средних (k-means), разделяющий исходную выборку объектов на k кластеров, при этом k сразу задается в начале работы. Разбиение достигается в результате минимизации целевой функции, заданной на множестве характеристик объектов. Наиболее часто используемая при этом функция – сумма квадратов отклонений координат всех объектов кластеров от центров соответствующих кластеров (сумма квадратов ошибок):

$$J(R) = \sum_{i=1}^k \sum_{j \in R_i} d(c_i, y_j), \quad (1)$$

где R_i - i -й кластер, c_i - центр i -го кластера, y_j - текущий объект, $d(c_i, y_j)$ - евклидово расстояние между центром i -го кластера и текущим объектом.

В методе кластеризации k-means используется евклидово расстояние, которое рассчитывается по формуле:

$$d(c_i, y_j) = \sqrt{\sum_{t=1}^n (c_{it} - y_{jt})^2}, \quad (2)$$

где y_j - текущий объект, n - размерность объектов, c_i – центр кластера (объект, максимально близкий к центру масс кластера).

Примером способа выбора количества кластеров в данном способе разбиения может служить график «каменистая осыпь» (Рис. 9):

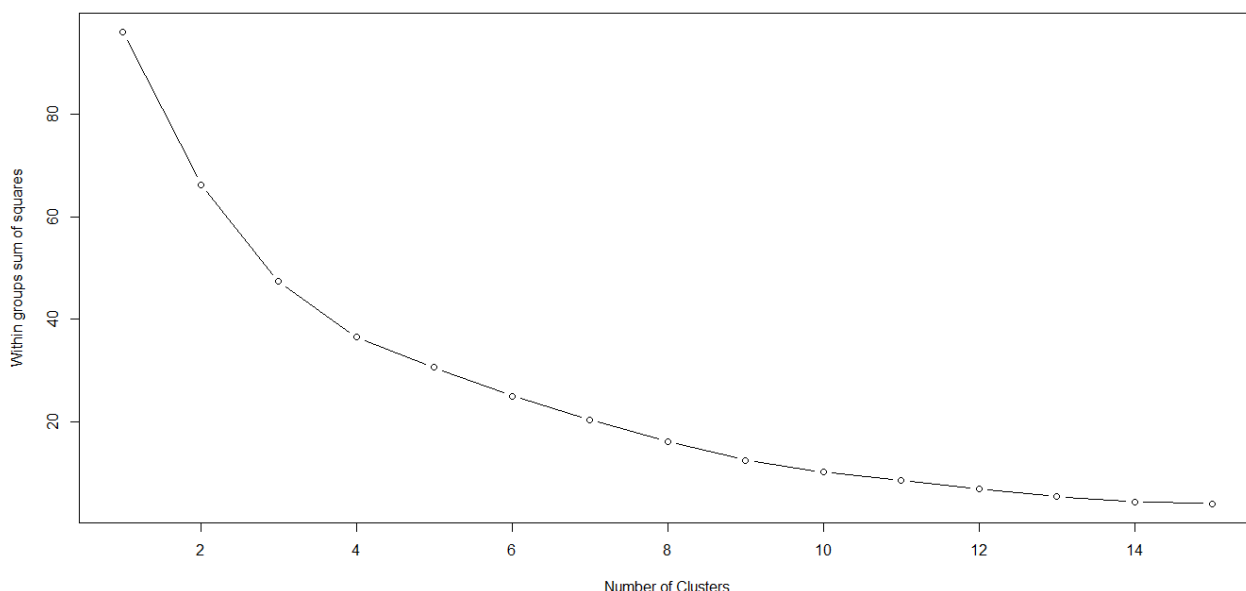


Рис. 9. График «каменистая осыпь» (пример)

На графике представлена зависимость суммы квадратов ошибок кластеризации от выбираемого количества кластеров. Наиболее оптимальным считается выбор такого количества кластеров, которое соответствует точке на графике, в которой виден наибольший «локоть» (перегиб). На данном примере это 3 или 4 кластера.

Первым шагом в алгоритме k-средних является разбиение на кластеры случайным образом. На каждом последующем шаге происходит пересчет центров тяжести кластеров и перераспределение объектов. При этом, на каждом шаге вычисляется сумма квадратов ошибок (сумма квадратов расстояний от объектов до центров кластеров), в общем случае, когда её улучшение становится уже незначительным, алгоритм останавливается (второй критерий остановки: выполнение максимального указанного количества итераций).

Среди преимуществ метода k-means можно выделить простоту использования и прозрачность алгоритма. Среди недостатков необходимость предварительного решения задачи по выбору количества кластеров, чувствительность алгоритма к выбору начальных центров кластеров, к выбросам и медленная работа с большими данными.

Самоорганизующиеся карты Кохонена

Одним из методов неиерархического кластерного анализа являются самоорганизующиеся карты Кохонена (SOM), предложенные как аналитический инструмент финским ученым Теуво Кохоненом в 1984 году [6].

Самоорганизующиеся карты Кохонена – мощный самообучающийся механизм кластеризации, позволяющий отобразить результаты в виде компактных и удобных для интерпретации двумерных карт [26].

Карта Кохонена – это тепловая карта, представляющая из себя тепловую сеть нейронов. Нейроны первого слоя соединены с нейронами второго слоя, расположенными в виде двумерной решетки. В первом слое находится столько же нейронов, сколько подаваемых на вход векторов. Во втором (выходном) слое число нейронов равно количеству кластеров (Рис. 10).

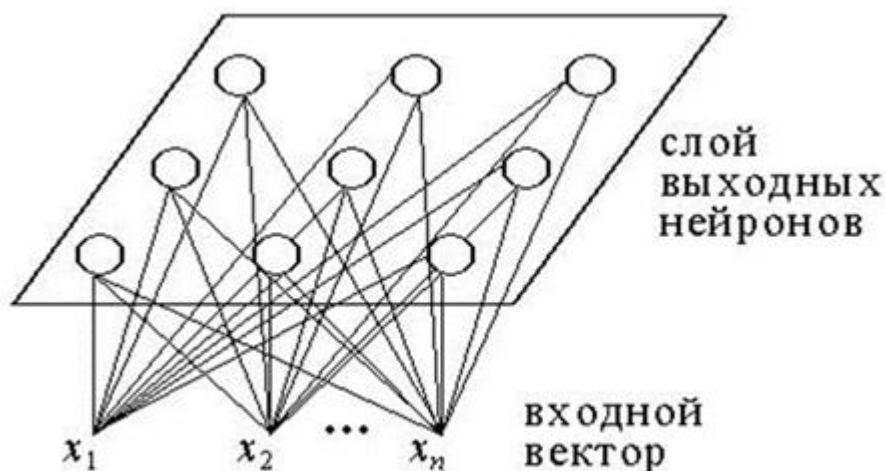


Рис. 10. Структура карты Кохонена

Перед началом обучения определяются начальные весовые коэффициенты нейронов. Способы их определения указаны ниже [27]:

- всем весам присваиваются малые случайные значения (на отрезке $[-1,1]$);
- в качестве начальных значений присваиваются значения случайно выбранных примеров из обучающей выборки;
- линейная инициализация (в этом случае веса инициализируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главных собственными векторами исходного набора данных).

Принцип обучения нейронов соревновательный и может быть описан фразой «победитель получает все». На каждом шаге обучения сети из исходного набора данных случайным образом выбирается вектор и осуществляется поиск нейрона выходного слоя, для которого расстояние между его вектором весов и выбранным вектором минимально. Расчет расстояния осуществляется метрикой Евклида:

$$\|X - Y\| = \sqrt{\rho(X, Y)} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}, \quad (3)$$

где $X, Y \subset R^n$, $x_i \in X$, $y_i \in Y$, $i = 1 \dots n$

Таким образом, если обозначить нейрон – победитель индексом r , то получится следующее равенство:

$$\|X_k - W_r\| = \min_{i=1 \dots m} \|X_k - W_i\|, \quad (4)$$

где $X_k, k = 1 \dots n$ - входные векторы (многомерные характеристики объектов), n - размерность входных векторов, $W_i, i = 1 \dots m$ - векторы весов нейронов выходного слоя, m - количество нейронов выходного слоя.

После выявления нейрона победителя с весами W_r , наиболее близкими входному вектору, корректируются веса нейрона-победителя и соседних с ним нейронов (соседство в рамках сети Кохонена). Корректировка происходит по следующему правилу:

$$W_i(q) = W_i(q-1) + \eta a_i(q-1)(X_k - W_i(q-1)), \quad (5)$$

где q – номер эпохи, η – коэффициент скорости обучения, $a_i(q)$ – функция соседства i -го нейрона с нейроном – победителем.

Обычно в качестве функции соседства предпочтение отдают Гауссовой [27], которая рассчитывается по формуле (6).

$$a_i(q) = \exp\left(-\frac{\rho(W_r, W_i)}{\sigma(q)}\right), \quad (6)$$

$$\text{Функция } a_i(q) \text{ является убывающей функцией от эпохи: } \sigma(q) = \frac{1}{\exp(q^{-2})}. \quad (7)$$

Когда изменения весов начинают меняться незначительно, алгоритм, в общем случае, останавливается (если не достигается другой критерий остановки, такой как: достижение выбранного количества эпох, либо отсутствие перехода векторов между кластерными элементами).

Результат работы алгоритма – тепловая многослойная карта, на которой объекты со сходными характеристиками находятся рядом, а более «теплым» оранжевым или красным цветом обозначены наиболее высокие значения рассматриваемого параметра. Пример карты в отечественной аналитической платформе Deductor на Рис. 11:

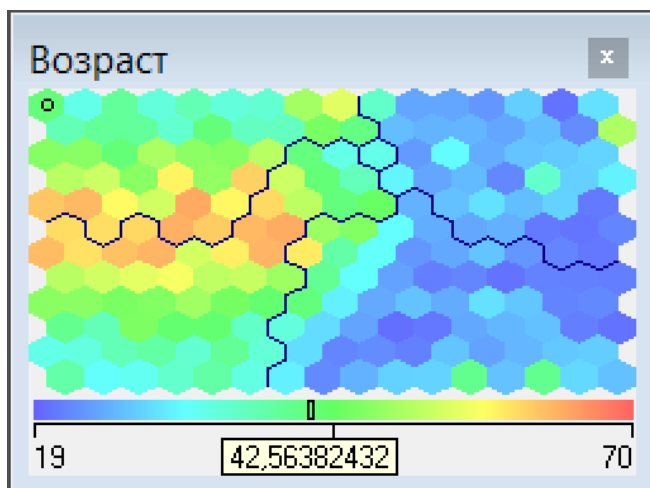


Рис. 11. Пример карты Кохонена в двумерном пространстве. Разбиение клиентов организации на возрастные группы.

Представление слоя карты Кохонена в трехмерном пространстве (Рис. 12):

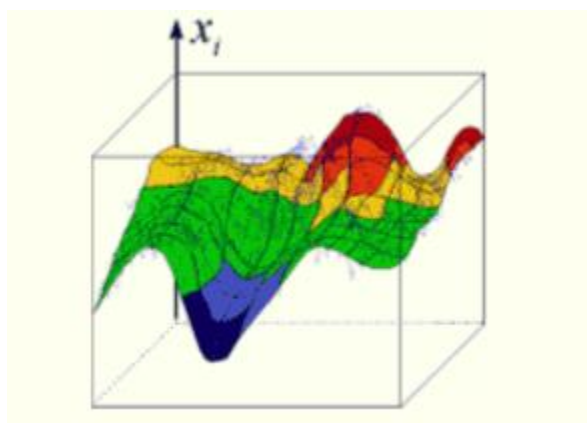


Рис. 12. Пример карты Кохонена в трехмерном пространстве.

2.2. Основы теории нечетких множеств

При работе с информацией не всегда данные являются полными, четко формулируются и определяются однозначно. В задачах плохо формализуемых, требующих субъективной человеческой оценки не в точных числовых значениях, а в терминах «достаточно хорошая стратегия» или «не очень хорошая стратегия», «зарботная плата ниже среднего» или «зарботная плата крайне низкая» удобно пользоваться инструментарием нечеткой логики, оперирующим экспертными понятиями, а не точными численными значениями. Нечеткость возникает при необходимости использования естественного языка и примерных значений, наиболее либо наименее вероятных событий, а не точных чисел.

Нечеткая логика – это расширение классических четких подходов. Основоположником теории нечетких множеств считается профессор Лотфи Заде, опубликовавший в 1965 году статью «Fuzzy sets» [25], в которой изложены основные идеи подходов к работе с плохо формализуемыми данными.

На практике применение нечетко-множественных подходов впервые осуществили студенты колледжа Королевы Мэри в Лондоне Е. Мамдани и С. Ассилиан. Ими был построен первый нечеткий контроллер для управления лабораторной моделью парового двигателя [23].

Одними из первых нечеткую логику стали применять финансисты, чьи задачи ежедневно требуют принятия сложных решений в условиях непредвиденного поведения рынка. Первый год использования системы FujiBank принес прибыль банку в размере \$770000 за месяц. Позже в нее стали инвестировать средства и промышленные гиганты США: Motorola, GeneralElectric, OtisElevator, PacificGas&Electric, Ford [7].

В настоящее время нечеткой логикой занимаются ученые разных профилей по всему миру. Например, Д. Антонелли и Д. Стадника описали в статье [22] алгоритм оценки решений по устранению ошибок, с целью выбора самого подходящего решения из перечисленных. В статье Е. Амрина и С. Рамадани [20] рассматривается применение системы нечеткого вывода в решении задачи об оценке работы заводов по производству цемента в Паданге, Индонезия. Статья Кузькина А. А. [11] описывает гибридную нейро-нечеткую систему анализа эффективности и результативности ИТ-процессов компании. Измерение успешности послепродажного сервисного обслуживания продуктов марки LG оценивали аналитики компании «Goldiran Customer Service», осуществляющей это обслуживание [24].

В экономическом анализе нечеткая логика применяется для обработки и интерпретации экспертных оценок, а также для построения экспертных систем.

Приведем формальное учебное определение [5] нечеткого множества:

Нечетким множеством A в некотором непустом пространстве X называется множество пар

$$A = \{(x, \mu_A(x); x \in X)\}, \quad (8)$$

где $\mu_A: X \rightarrow [0, 1]$ – функция принадлежности нечёткого множества A (характеристическая функция) [4]. $\mu_A(x_i)$ показывает степень принадлежности i -го объекта данных множеству (кластеру) A и может колебаться от 0 до 1. Если бы функция принадлежности могла принимать одно из двух значений: 0 или 1, то множество A было бы четким. То есть четкие множества являются подмножеством нечетких.

Одной из характеристик нечеткого множества является понятие *носителя*. Носителем нечеткого множества называется множество, содержащее те и только те элементы базового пространства нечёткого множества, для которых значения функции принадлежности этого нечеткого множества отличны от нуля. Т.е. носитель – множество элементов хотя бы на сколько-то принадлежащих рассматриваемому нечеткому множеству. На Рис. 13 носителем является множество $[100, 700]$, а само нечеткое число описывает значение «примерно 400». По оси X откладываются значения, по оси Y – степень принадлежности, или же вероятность, с которой мы можем отнести фактическое значение к числу 400.

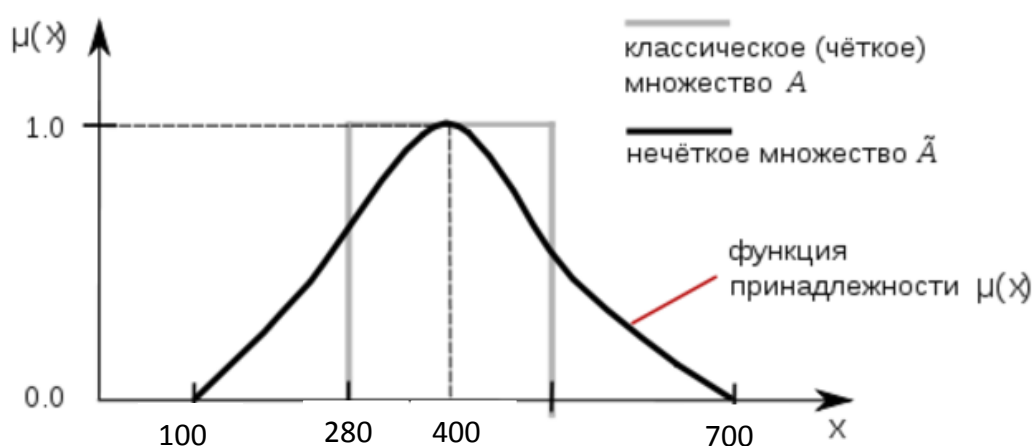


Рис. 13. Нечёткое множество и классическое чёткое [36].

Наиболее часто встречаются функции принадлежности, имеющие треугольную или трапецевидную форму, так как они дают возможность технически просто производить математические операции с нечеткими множествами и при этом достаточно хорошо описывают рассматриваемые объекты. Формула для задания треугольной функции принадлежности выглядит следующим образом:

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (9)$$

В приложении 1 в таблице 1 представлены другие возможные представления функции принадлежности, в скобках рядом с названиями для примера указаны соответствующие функции в прикладном пакете MATLAB, формирующие нечеткое число.

Арифметические действия над нечеткими числами производятся в соответствии с классической теорией множеств [5, с. 298], основные операции над нечеткими числами отражены в приложении 1 в таблице 2.

Важную роль в нечеткой логике играет термин «лингвистическая переменная». По смыслу лингвистическая переменная – это тот критерий, которым мы описываем тот или иной объект или явление при помощи нечеткого множества. Значениями лингвистической переменной являются слова (термы), в отличие от привычных числовых значений обычных переменных [33]. Формальное определение лингвистической переменной звучит следующим образом [5]:

Лингвистической переменной называют пять объектов: $(x, T(x), U, G, M)$, где x - собственное имя переменной; $T(x)$ - терминальное множество, т.е. набор значений переменной (нечетких меток); U - множество объектов (или универсум); G - синтаксические правила употребления; M - семантические правила употребления.

Таким образом, лингвистическая переменная, описывающая возраст человека, будет выглядеть следующим образом:

- x - «возраст»;
- $T(x)$ - множество нечетких переменных, таких как «молодой», «взрослый», «старый».

Каждая нечеткая переменная определяется функций принадлежности;

- U - множество целых чисел от 0 до 100;
- G - «сильно», «не сильно». Такие правила употребления позволят образовать новые значения, например: «сильно старый», «не очень взрослый»;
- M - математические правила, которые определяют вид функции принадлежности для каждого нового значения, полученного при помощи синтаксического правила G .

При формировании нечеткой экспертной системы помимо лингвистических переменных необходимо задать базу правил, определяющую, как входные переменные будут влиять друг на друга, и как они будут влиять на переменную выхода. База правил строится по типу логического выражения «если» ..., «то» ...:

$$\begin{aligned} & \text{Если } X \text{ это } A_1 \text{ и } Y \text{ это } B_1 \text{ то } Z \text{ это } C_1, \\ & \text{Если } X \text{ это } A_2 \text{ и } Y \text{ это } B_2 \text{ то } Z \text{ это } C_2, \\ & \dots\dots\dots \\ & \text{Если } X \text{ это } A_n \text{ и } Y \text{ это } B_n \text{ то } Z \text{ это } C_n, \end{aligned} \tag{10}$$

где X, Y – множества входных переменных, Z – множество выходных переменных, $A_i, B_i, C_i, i = 1..n$ – нечеткие множества.

После задания переменных, функций принадлежности, описывающих переменные, базы правил становится возможным определить результат - выходную переменную.

Приведем пример задания всех перечисленных параметров и определения результата.

Цель описываемой экспертной системы – определение уровня кредитоспособности заемщика.

Факторы (входные переменные), оказывающие влияние на уровень кредитоспособности заемщика:

- опыт работы;
- ежемесячный доход;
- сумма текущих обязательств.

Указанные факторы, влияющие на уровень кредитоспособности, могут быть описаны нечеткими множествами, например, опыт работы – «средний», ежемесячный доход – «малый», сумма текущих обязательств – «низкая». Для каждого нечеткого множества теперь зададим функцию принадлежности, которая в свою очередь определит, чем измеряется «средний» опыт работы, и чем опыт работы «большой». Функции принадлежности для нечетких множеств настраиваются на основании мнений экспертов.

Таким образом, база правил может быть следующей:

- Если (опыт работы - «большой») И (ежемесячный доход – «высокий») И (сумма текущих обязательств – «низкая»), ТО (уровень кредитоспособности – «высокий»);
- Если (опыт работы – «средний») И (ежемесячный доход – «средний») И (сумма текущих обязательств – «высокая»), ТО (уровень кредитоспособности – низкий), и т.д.

На основании такой экспертной системы можно определить уровень кредитоспособности для нового заемщика, исходя из его характеристик: опыта работы, ежемесячного дохода, суммы текущих обязательств.

Аналогично работает нечеткая экспертная система прогноза уровня риска развития профессионально обусловленных заболеваний [14], экспертная система для оценки технического состояния двигателя на основании значений параметров рабочего процесса и топливоподачи [15], система оценивания эффективности инвестиционных проектов на основании таких входных параметров, как IRR, NPV, PB [17] и другие.

В данной работе будет построена подобная экспертная система, оценивающая эффективность работы сотрудников компании.

2.3. Использование искусственных нейронных сетей

Перечень задач, решаемых при помощи искусственных нейронных сетей довольно широк: это задачи и классификации, и кластеризации, и построения прогноза.

Нейронные сети начали изучать с середины XX века, поскольку перспектива совмещения логики человеческого мышления со скоростью работы компьютера была и остается многообещающей. За время изучения ИНС было написано множество статей, посвященных решению разных задач из разных областей знаний. К примеру, авторами статьи из журнала SCIENCE AND BUSINESS: DEVELOPMENT WAYS [8] проиллюстрирована возможность эффективного применения нейронных сетей в сфере финансовой аналитики для построения динамических моделей курсов ценных бумаг, прогнозирования стоимости валюты. В статье И.И. Слеповичева [18] описано решение задачи классификации нейронными сетями при обнаружении DDoS атак.

Нейронная сеть – это структура, состоящая из одного или нескольких слоев. В каждом слое может находиться один или несколько нейронов (их количество может меняться от слоя к слою).

Один нейрон человека (Рис. 14) состоит из тела (сомы), дендритов (принимающих информацию от аксонов других нейронов) и аксонов (передающих информацию дендритам других нейронов). Передача сигнала от одного нейрона к другому осуществляется посредством синапсов, находящихся на стыке дендрита с аксоном [1]. Все синапсы обладают разными «весами».

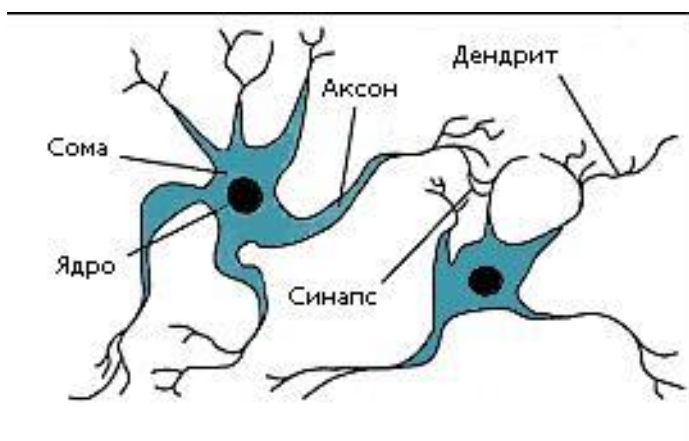


Рис. 14. Два биологических нейрона. Иллюстрация передачи сигнала

Входные сигналы, попадающие на вход нейрона стремятся возбудить или затормозить его. Если возбуждение нейрона численно превышает некоторый порог, то полученный

сигнал проходит дальше через аксон рассматриваемого нейрона к дендритам других нейронов и так же распространяется (или не распространяется) далее.

Структурно один нейрон может быть представлен следующим образом (Рис. 15):

1. Набор связей (синапсов), каждый из которых характеризуется своим весом (силой): сигнал x_j на входе синапса j , связанного с нейроном k , умножается на вес w_{kj} . Вес искусственного нейрона принимает как положительные значения, так и отрицательные.
2. Сумматорная функция Σ складывает сигналы, которые подаются ему на вход.
3. Функция активации $\varphi(\cdot)$ ограничивает амплитуду выходного сигнала нейрона
4. Пороговый элемент b_k , использование которого обеспечивает эффект аффинного преобразования выхода сумматора [4, с. 43].

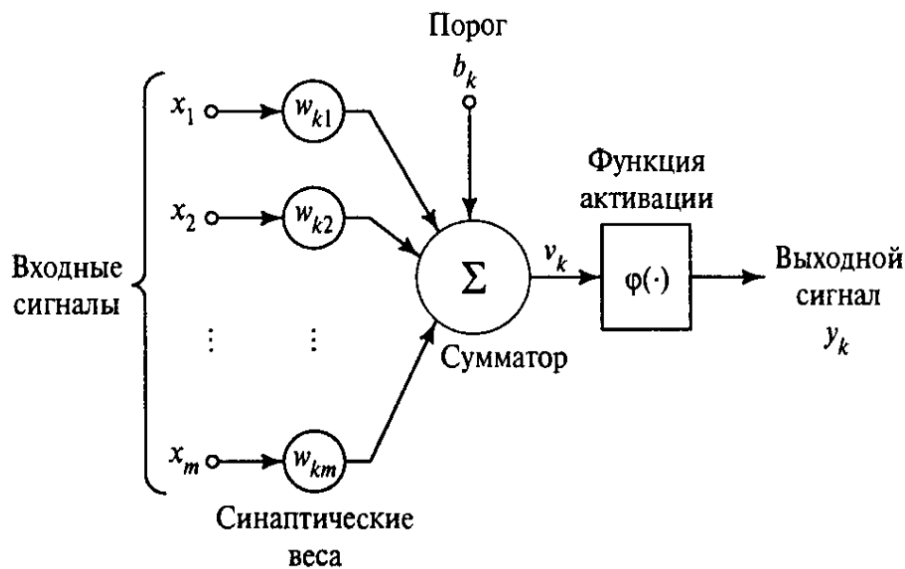


Рис. 15. Принцип работы искусственного нейрона

Математически нейрон описывается в литературе следующей системой уравнений [3]:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (11)$$

$$y_k = \varphi(u_k + b_k)$$

где x_1, \dots, x_m – входные сигналы; w_{k1}, \dots, w_{km} – синаптические веса нейрона k ; u_k – линейная комбинация входных воздействий; b_k – порог; $\varphi(\cdot)$ – функция активации; y_k – выходной сигнал нейрона.

Функцией активации $\varphi(\cdot)$ может быть пороговая (формула 13), кусочно-линейная (формула 14) или сигмоидальная (формула 15) функция (аналоги в прикладном пакете MATLAB: hardlim, purelin, logsig соответственно):

$$\varphi(v) = \begin{cases} 1, & \text{если } v \geq 0; \\ 0, & \text{если } v \leq 0. \end{cases} \quad (12)$$

$$\varphi(v) = \begin{cases} 1, & \text{если } v \geq \frac{1}{2}; \\ |v|, & \text{если } \frac{1}{2} > v > -\frac{1}{2}; \\ 0, & \text{если } v \leq -\frac{1}{2}. \end{cases} \quad (13)$$

$$\varphi(v) = \frac{1}{1 + \exp(-av)}, \quad (7)$$

где a - параметр наклона сигмоидальной функции, изменяя который можно построить функции с различной крутизной.

Основными архитектурами ИНС являются [3]: однослойная сеть прямого распространения (Рис. 16), многослойная сеть прямого распространения (Рис. 17), рекуррентные сети (Рис. 18).

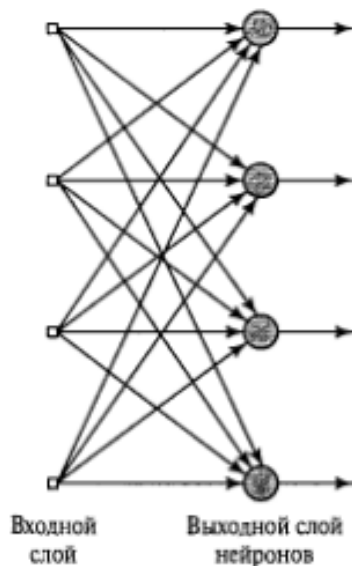


Рис. 16. Однослойная сеть прямого распространения

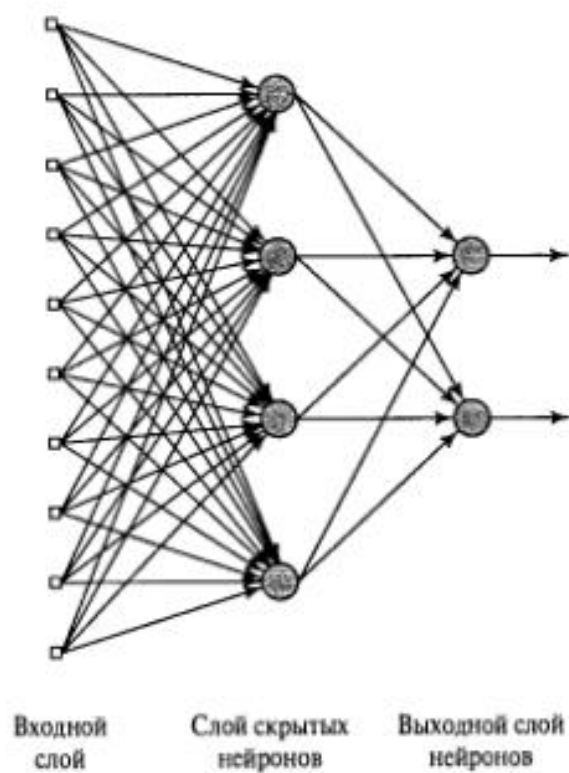
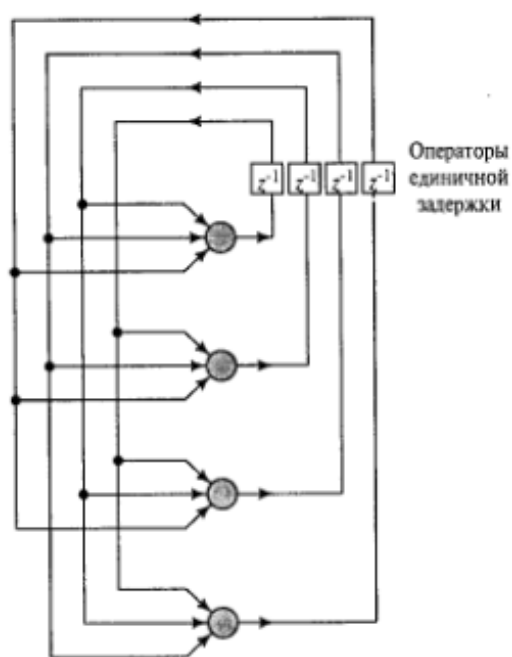
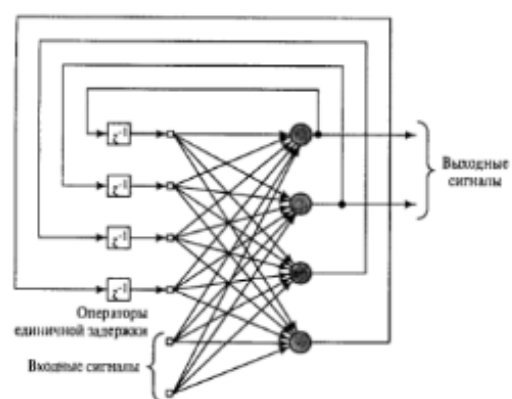


Рис. 17. Многослойная сеть прямого распространения



а)



б)

Рис. 18. Рекуррентные сети: а) Однослойная сеть; б) Многослойная сеть

В сетях прямого распространения информация поступает в нейроны входного слоя и передается в нейроны выходного слоя. В рекуррентных сетях указанные процессы сопровождаются обратной связью от выходного слоя к входному.

Подобно нейронной сети человеческого мозга, для того, чтобы выдать адекватный результат касательно решаемой задачи, искусственную нейронную сеть необходимо обучить. Цель обучения – скорректировать весовые коэффициенты синапсов. Если рассматривать обучение «с учителем», то весовые коэффициенты на каждом шаге обучения изменяются таким образом, чтобы полученный нейронной сетью выход был похож на «правильный» или желаемый результат. Обучение «с учителем» - это случай, когда есть данные, с которыми мы можем сравнить выход ИНС, альтернативная задача – «обучение без учителя», подразумевает, что нам не с чем сравнить результат обучения ИНС.

Вся имеющаяся для обучения выборка делится на три части, на которых и завязан весь процесс работы этой ИНС. Сеть обучается на самом большом, тренировочном (обучающем) множестве, на второй части данных проходит валидацию, и на третьей части – на тренировочном множестве – демонстрирует качество обучения.

В данной работе нейронная сеть могла быть применена, например, для выявления зависимости между качеством работы сотрудника и характеристиками рассматриваемого сотрудника.

ГЛАВА 3. ПРОЕКТИРОВАНИЕ ПРОТОТИПА БАЗЫ ДАННЫХ В MS ACCESS

3.1. Описание ограничений предметной области

Производственная компания, выпускающая технически сложный продукт, является коммерческой организацией, направленной на получение прибыли. Для стабильной работы производства необходимо своевременно заказывать у поставщиков детали для производства продукта. Детали заказываются как у локальных поставщиков, находящихся на расстоянии нескольких часов езды на машине, так и у глобальных, осуществляющих доставку преимущественно морем (длительность доставки может достигать нескольких месяцев) и в крайнем случае авиаперевозкой. Закупленные детали хранятся в определенной последовательности на самом производстве, готовые для сборки, а также без соблюдения строгой последовательности на складе. Если на складе деталей слишком мало, то ситуация расценивается как рискованная, если слишком много – загрузка складских помещений не оптимальна. Для регулирования стабильных поставок и бесперебойной работы производства в компании существует отдел по работе с поставщиками, контролирующий заказ и доставку деталей на производство и на склад.

Менеджеры отделов и менеджеры групп ежегодно ставят себе и своим подчиненным цели (а также получают их от руководства), измеряемые таргетами, разбивают их на стратегии, измеряемые контрольными элементами, и достижение каждого контрольного элемента делегируют сотрудникам, находящимся ниже в иерархии. При этом стратегия делегируется в цель сотрудника, а контрольный элемент – в таргет. Информацию о результатах работы сотрудников отдела по работе с поставщиками предполагается хранить в единой базе.

Представляя в виде наглядной схемы, у сотрудника есть (на один финансовый год) следующие цели и задачи (Рис. 19):

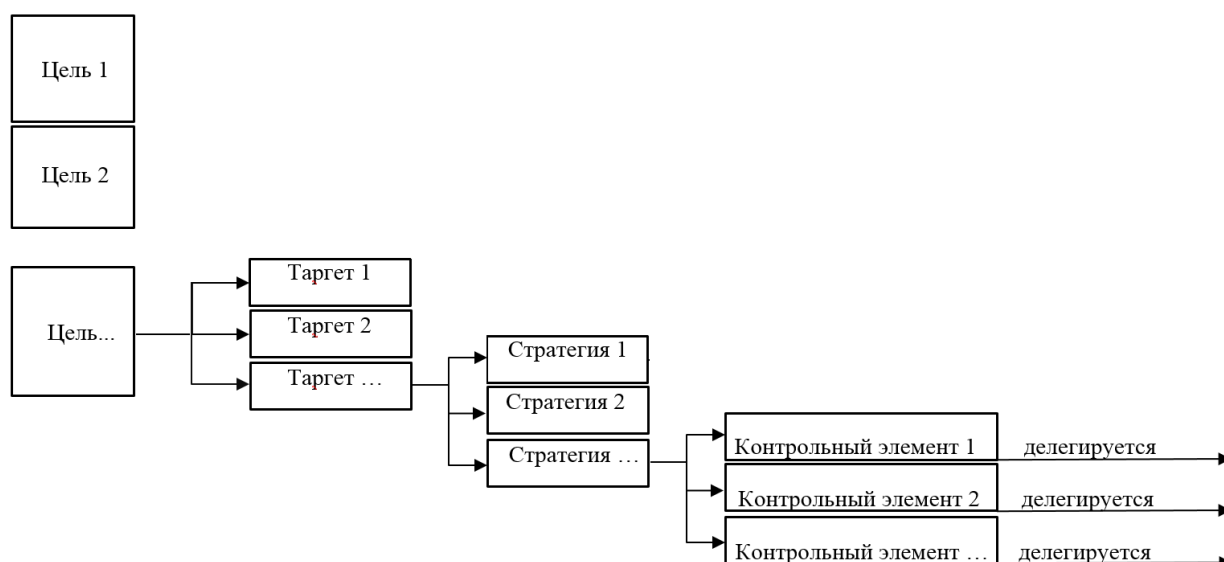


Рис. 19. Схема пула целей сотрудника.

И ежемесячно (поквартально) сотрудник вносит фактический результат таргета за прошедший период.

Пользователями базы данных будут: менеджер отдела, менеджеры групп, сотрудники, непосредственно работающие с поставщиками.

3.2. Определение требований к системе через информационные потребности пользователей

Проектируется база данных для хранения информации о годовых целях сотрудников, разбитых на стратегии, а также хранения ежемесячных либо поквартальных фактических результатов работы сотрудников отдела закупок. При этом информационные потребности пользователей можно представить в следующем виде:

Менеджеры	Просмотр целей всех сотрудников, фактических результатов достижения целей; Постановка собственных целей; Получение целей, делегированных руководством; Делегирование стратегий в цели подчиненных
Сотрудники	Просмотр собственных целей, (как делегированных руководством, так и поставленных лично), заполнение фактических результатов достижения целей

Таблица 1. Потребности пользователей.

Таким образом, к разрабатываемой базе предъявляются следующие требования:

- База должна хранить цели и стратегии (аналог задач) сотрудников, измеряемые таргетами и контрольными элементами соответственно;
- Каждая цель имеет помимо описания категорию, выбираемую из справочника; информацию о том, кем и когда она была ему делегирована, если она не была создана самим сотрудником; может быть описана несколькими характеристиками, для которых также должен существовать отдельный справочник; имеет таргеты, разбивается на стратегии;
- Каждый таргет характеризуется его описанием; если таргет был делегирован, информацией о том, кем при делегировании был назначен владелец данного таргета (ответственным или поддерживающим); таймингом (как часто меряем показатель – ежемесячно или поквартально); единицами измерения в которых производится измерение; годом, к которому относится; целевым значением (target value); минимальным обязательным для достижения значением (commitment value);
- Каждая стратегия имеет собственное описание и один или несколько контрольных элементов;
- Каждый контрольный элемент характеризуется описанием, таймингом (как часто меряем показатель – ежемесячно или поквартально), единицами измерения в которых производится измерение, годом, целевым значением, минимальным обязательным для достижения значением (аналогично таргету), а также ответственными и поддерживающими сотрудниками, которым эта стратегия делегируется. При этом, один и тот же сотрудник не может одновременно быть ответственным и поддерживающим для одного контрольного показателя;
- Цели и таргеты должны относиться к сотруднику, заполняющему свой профиль в базе, стратегии и контрольные элементы – к тем сотрудникам, которым он делегирует разбитую на стратегии (задачи) цель;
- С указанной периодичностью сотрудник должен иметь возможность в удобной форме вносить фактические результаты своей работы в базу;
- Должна поддерживаться возможность делегирования стратегий («каскадирования») ответственным за них и тем, кто должен выступать в роли поддерживающих достижение стратегии.

Входная информация представлена в таблице с объектами базы данных и их описанием в приложении 2.

3.3. Реализация проекта в среде СУБД MS ACCESS, включая создание пользовательского приложения

В соответствие с ИЛМ данных в СУБД MS ACCESS была создана схема разрабатываемой базы данных (Рис. 20).

При создании связей между таблицами была обеспечена целостность данных, установлено каскадное обновление полей и удаление связанных записей.

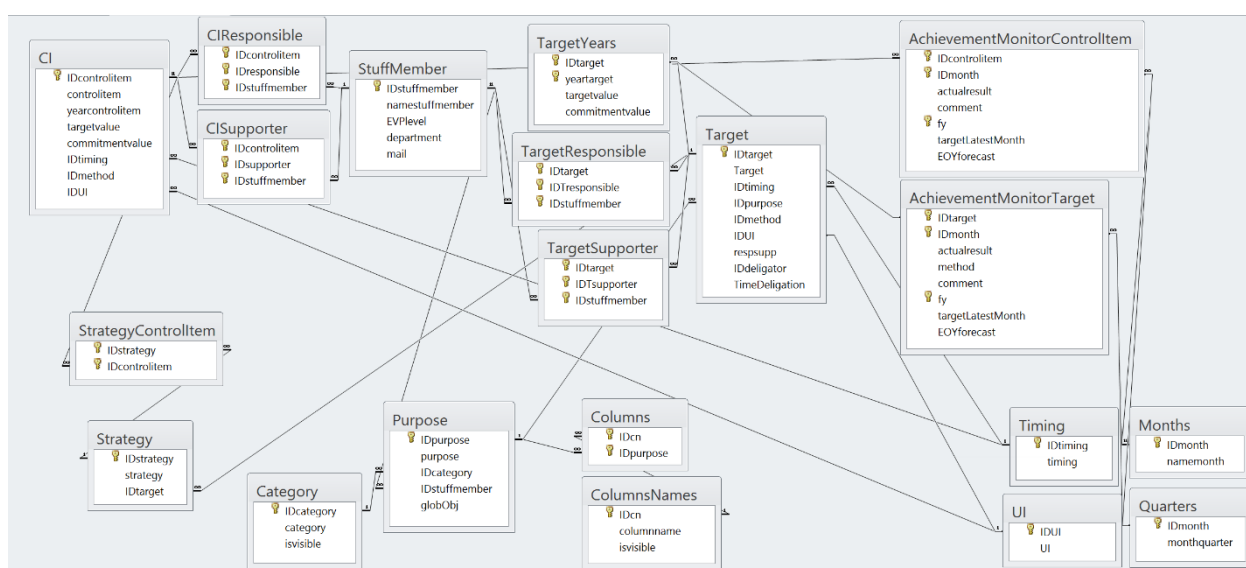


Рис. 20. Схема данных в MS ACCESS.

Внесение данных пользователем осуществляется посредством форм, представленных поэтапно в приложении 3, в данном разделе приведем для примера несколько изображений, иллюстрирующих внесение пользователем целевых значений таргета и стратегии на предстоящий год (Рис. 21, 22):

Targets of the objective: Reduce mislabeling

Back

Look at other your targets

Target: Number of parts with mislabeling UI %

Timing: monthly

Method: EOY

Year target	Target Value	commitmentvalue
2018	0,00	10,00
*	0,00	0,00

Deligated by: Nikitin Andrey

TimeDelegation: 31.03.2018 18:46:03

You are: Supporter

Запись: 1 из 1

Нет фильтра

Поиск

Рис. 21. Внесение целевых значений таргета.

Strategies for objective: Reduce mislabeling

Back

Look at other your strategies

Choose Target: Number of mislabeling parts, that are not in the work

Strategy: Plan the time effectively

Controlitem: spend on 1 vendor minimum time daily	UI: minutes	Year: 2017	Responsible
Timing: monthly		Target value: 15,00	
Method: EOY		Commitment: 30,00	Supporters
Controlitem:	UI:	Year:	Responsible
Timing:		Target value:	
Method:		Commitment:	Supporters

Рис. 22. Внесение целевых значений контрольных элементов в стратегии.

ГЛАВА 4. ПОСТРОЕНИЕ СИСТЕМЫ НЕЧЕТКОГО ВЫВОДА СРЕДСТВАМИ MATLAB

4.1. Связь базы и системы нечеткого вывода

База данных создается для централизованного сбора данных о результатах работы сотрудников. Чтобы эту работу оценить, менеджеры проводят анализ, в частности, по отчетам и графикам. Таким образом, база данных агрегирует информацию, а экспертная система помогает провести последующий анализ собранной информации.

Для обоснования выбора метода проведения анализа представим: пример отчета для менеджеров, данные в котором берутся непосредственно из базы и проходят минимальную обработку, и прообраз отчета, основанного на результатах создаваемой экспертной системы. Для простоты рассмотрим только некоторые переменные (1 блок). В первом отчете за каждый достигнутый критерий сотрудник получает 1 балл, за каждый недостигнутый – 0. Во втором отчете связь переменных и результирующего балла определяется правилами, заданными в экспертной системе.

Разница между двумя отчетами в том, что первый учитывает в многокритериальном ранжировании лишь то, достигнута цель или нет (Рис. 23, Рис. 25), а второй учитывает то, насколько именно эта цель была достигнута, и насколько сложной была сама работа (Рис. 24, Рис. 26).

Name	Wrong Marking		
	Quantity	NotInTheWork	NoExplanation
Artur Taliev	0.04	0.29	0.00
Eena Kononova	0.12	0.19	0.00
Anna Masalskaya	0.06	0.17	0.67
Irina Germanova	0.07	0.00	0.00
Denis Matveev	0.04	0.30	0.30
Maya Lishneva	0.05	0.27	0.07
Fedor Krasilnikov	0.06	0.71	0.14
Anastasiya Babenko	0.08	0.88	0.13
Mila Feoktistova	0.01	0.67	0.33
David Chumertiv	0.40	0.00	0.00
Leonid Losev	0.02	1.00	1.00

Рис. 23. Отчет для менеджеров, вариант 1

Name	Wrong Marking			Difficulty
	Quantity	NotInTheWork	NoExplanation	
Artur Taliev	0,04	0,29	0,00	3
Eena Kononova	0,12	0,19	0,00	7
Anna Masalskaya	0,06	0,17	0,67	9
Irina Germanova	0,07	0,00	0,00	5
Denis Matveev	0,04	0,30	0,30	8
Maya Lishneva	0,05	0,27	0,07	4
Fedor Krasilnikov	0,06	0,71	0,14	3
Anastasiya Babenko	0,08	0,88	0,13	5
Mila Feoktistova	0,01	0,67	0,33	4
David Chumertiv	0,40	0,00	0,00	9
Leonid Losev	0,02	1,00	1,00	3

Рис. 24. Отчет для менеджеров, вариант 2

В данном примере для простоты понимания и наглядности в ранжирование было включено три критерия: количество случаев неверной маркировки деталей, количество необработанных сотрудником проблем и количество не поступивших объяснений от поставщика (Рис. 23, Рис. 25), исходя из этого в варианте 1 оценка трехбалльная. Логично, что чем показатель ниже, тем лучше. Цели по показателям в данном примере: 0.5, 0, 0 соответственно. При этом для нечеткой системы был добавлен критерий сложности работы с поставщиком, Difficulty (Рис. 24, Рис. 26), измеряемый по десятибалльной шкале. В этом случае результирующая оценка также измерялась по десятибалльной шкале.



Рис. 25. Значения для отчета вариант 1

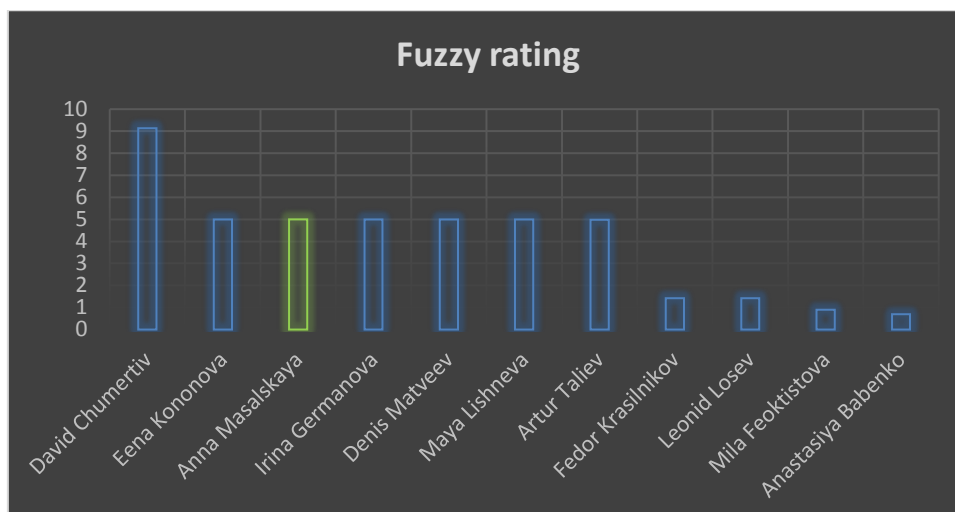


Рис. 26. Значения для отчета вариант 2

Рассмотрим пример Анны Масальской. В простом ранжировании она не достигла ни одной цели, поэтому в рейтинге оказалась в последней тройке. Однако при учете степени достижения целей, а главное сложности работы оказалась третьей сначала (поскольку сложность ее по 10 балльной шкале равна 9). Такая ситуация – яркий пример недоучета всех факторов работы сотрудников, а также приложенных ими усилий по каждому из факторов, если компания не использует фактически важные показатели, численно измерить которые нельзя.

Таким образом, отчет – это результат всего анализа. Данные для отчета – это результат работы экспертной системы. Входные данные для экспертной системы – это собранная в базе данных информация, то есть результаты работы сотрудника. Обмен данными между базой, экспертной системой и финальным отчетом в данной работе осуществляется вручную через графические окна, однако может быть автоматизирован при помощи написания кода. Пример кода для ввода данных в уже построенную модель Simulink представлен в приложении 5. В приложении рассматривается пример одного сотрудника, соответственно, на вход подается одна строка данных. Аналогично можно дописать код до загрузки данных не по одному, а по нескольким сотрудникам, где будет несколько строк входа. Осуществлять загрузку входных данных в таком случае можно будет уже из файла, импортируемого по запросу из базы данных Microsoft Access.

Вывод финального отчета в автоматическом режиме (либо по кнопке в базе данных Access) программируется при помощи макросов на языке VBA в файле Excel (либо в самой базе данных).

4.2. Определение лингвистических переменных для системы нечеткого вывода и создание блоков переменных

Лингвистические переменные являются критериями оценки сотрудника и входными переменными проектируемой экспертной системы. Каждая переменная – это параметр, который определяет степень достижения KPI сотрудником, и который состоит из нескольких функций, описывающих нечеткие числа. Некоторые переменные могут относиться к одной тематике, что делает логичным объединение их в отдельный блок. В данной работе были определены лингвистические переменные и разбиты на блоки следующим образом:

Блок WRONG MARKING

1. Quantity (виды деталей (количество) с неверной маркировкой на коробках, не отраженные в системе; считается в процентах от всех видов деталей сотрудника, имеющих неверную маркировку);
2. NotInTheWork (количество видов деталей с неверной маркировкой, по которым сотрудник не запросил объяснений и контрмер от поставщика; так же в процентах);
3. NoExplanation (количество не пришедших от поставщика писем с объяснениями и планами по контрмерам).
4. Экспертная оценка: DifficultVendors (количество поставщиков, с которыми сложно работать; в баллах от 1 до 10)

Блок RISK PARTS

1. HighRiskPartQuantity (количество видов деталей, по которым высок риск опоздания поставки из-за несвоевременного их прибытия (например, поставщик оповестил о поломке машины, или его надолго задерживают на таможне); в процентах от всех видов деталей);
2. HighRiskPartInTheWorks (количество высокорисковых видов деталей, с которыми сотрудник начал работу по снижению риска; в процентах от всех высокорисковых);

Блок SELF-DEVELOPMENT

1. TrainingsQuantityListener (количество посещенных курсов внутри организации в качестве слушателя);
2. TrainingsQuantitySpeaker (количество посещенных курсов внутри организации в качестве спикера);

3. Экспертная оценка: Proactivity (участие в смежных проектах на работе, не указанных в трудовом договоре, оценка вовлеченности человека, его проактивности; измеряется в количестве дополнительных проектов).

Блок SAFETY-DAYS

1. SafetyDays1 (срок, на который хватает запаса деталей; среднее количество дней по видам деталей, когда дней меньше 10);
2. SafetyDays2 (срок, на который хватает запаса деталей; среднее количество дней по видам деталей, когда дней больше 10).

Итого, всего 11 критериев оценки.

Стоит пояснить разделение блока SAFETY-DAYS на две переменных: так как берется среднее значение дней, то при использовании одной переменной слишком высокие значения могли компенсировать критично низкие, таким образом дав итоговую оценку в пределах нормы, игнорируя фактическую проблему.

4.3. Описание работы экспертной системы в пакете MATLAB.

Все выделенные переменные были собраны в блоки, где каждый блок имел на выходе определенный результат. Все блоки также были связаны между собой в единую модель при помощи MATLAB Simulink (Рис. 27). Исходные 4 блока разделили также на блоки: отвечающий за саморазвитие и результаты непосредственной работы. Конечный результат определяется сверткой: результат по блоку саморазвития с коэффициентом 0.35 суммировался с результатом по блоку итогов работы с коэффициентом 0.65.

Подробное описание использованных переменных, описанных в пакете MATLAB, а также правил, заданных для работы блоков, можно найти в приложении 4.

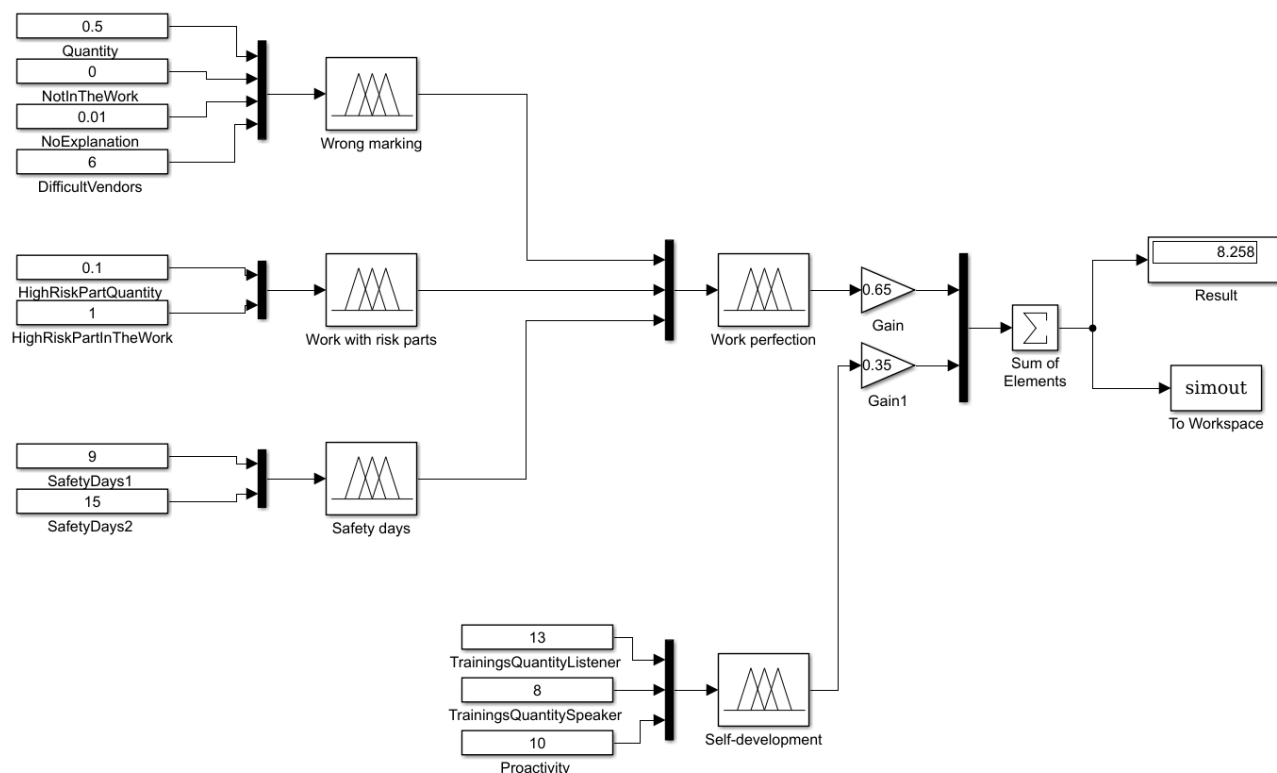


Рис. 27. Система нечеткого вывода

Результат работы системы рассмотрели на нескольких примерах (Табл. 1). Описание используемых переменных приводилось в предыдущем параграфе.

	WRONG MARKING				RISK PARTS		SAFETY-DAYS		SELF-DEVELOPMENT			Result
	Quantity	NotInTheWork	NoExplanation	DifficultVendors	Quantity	InTheWorks	SafetyDays1	SafetyDays2	TrainingsQuantityListener	TrainingsQuantitySpeaker	Proactivity	
1	0.3	0.33	0.11	5.00	0.04	0.55	1.00	21.00	0.00	3.00	1.00	3.24
2	0.6	0.41	0.53	5.00	0.08	0.23	2.00	36.00	9.00	9.00	10.00	4.47
3	0.1	0.17	0.44	2.00	0.75	0.10	6.00	16.00	0.00	7.00	0.00	5.52
4	0.2	0.81	0.46	2.00	0.11	0.08	7.00	14.00	0.00	5.00	8.00	5.30
5	0.15	0.64	0.66	6.00	0.94	0.50	10.00	32.00	11.00	5.00	9.00	4.14
6	0.17	0.29	0.88	5.00	0.91	0.01	5.00	29.00	7.00	6.00	8.00	5.69
7	0.31	0.26	0.30	4.00	0.10	0.30	2.00	31.00	5.00	0.00	3.00	4.30
8	0.24	0.31	0.98	0.00	0.15	0.92	7.00	20.00	2.00	2.00	3.00	3.87
9	0.12	0.47	0.21	5.00	0.99	0.21	6.00	13.00	9.00	1.00	1.00	4.41
10	0.28	0.70	0.52	9.00	0.31	0.49	1.00	23.00	15.00	1.00	10.00	3.28
11	0.5	0.00	0.01	6.00	0.10	1.00	9.00	15.00	13.00	8.00	10.00	8.26

Табл. 1. Данные для системы нечеткого вывода

По результатам работы системы видим, что сотрудник под номером 11 имеет наивысший балл, в то время как 1 и 10 самые низкие оценки. Если посмотреть на фактические значения по каждой переменной для этих сотрудников нетрудно убедиться в корректности результата. У сотрудника под номером 11 самые высокие баллы по блоку саморазвитие и одни из лучших показатели по работе с неверной маркировкой, высокорисковыми деталями, а также критерий запаса деталей в днях. В то время как у сотрудника 10 среднего запаса деталей (переменная SafetyDays1) хватает на 1 день, почти

половина высокорисковых деталей даже не в работе, и 70% деталей с неверной маркировкой также остались необработанными. Опять же, сложность поставщиков, с которыми работает сотрудник 10 равна 9. Этот факт дает задуматься о том, что имеет место более успешному сотруднику давать более сложных поставщиков, а у работника, не справляющегося с текущим объемом заданий, оставить поставщиков, с которыми работать проще.

ЗАКЛЮЧЕНИЕ

Для успешной работы любой компании необходимо правильно оценивать работу сотрудников, так как результат этой оценки способен помочь правильно распределить нагрузку, определить претендентов на повышение (или на дополнительное обучение) и в целом показать эффективность использования кадровых ресурсов. А для того, чтобы иметь возможность оценивать результаты работы, необходимо иметь доступ к фактическим показателям достижения целей. Для того, чтобы сделать все вышеперечисленное возможным, в данной работе была поставлена и выполнена цель: спроектировать базу данных для централизованного хранения целевых показателей на год, а также создать систему нечеткого вывода для получения итоговых балльных оценок эффективности работы сотрудников. Поэтапно были выполнены следующие задачи:

1. Определены информационные потребности пользователей проектируемой БД, сформулированы требования и ограничения предметной области;
2. Создан прототип базы данных для хранения показателей;
3. Определены критерии, по которым будет производиться оценка эффективности сотрудников, после чего сгруппированы по блокам;
4. В MATLAB Simulink построена система нечеткого вывода показателей эффективности работы сотрудников на основе выделенных критериев;
5. Работа системы протестирована на данных, собранных в базе, результатом является ряд балльных оценок 11-и сотрудников по выделенным критериям.

Использование предложенных в работе методов оценки эффективности работы сотрудников может быть адаптировано под любой другой отдел или компанию. При этом набор критериев, их описание, а также база правил, по которым работает система, должны быть скорректированы.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Минько Э. Оценка эффективности коммерческих проектов: Учебное пособие/ Минько Э., Завьялов О., Минько А. – СПб.: Питер, 2014. — 368с.
2. Тихонов Э.Е. Методы прогнозирования в условиях рынка: учебное пособие/ Тихонов Э.Е. - Невинномысск, 2006. — 221 с.
3. Хайкин С. Нейронные сети. Полный курс 2-е изд. Пер. с англ./ Хайкин С. – М.: Издательский дом "Вильямс", 2006. — 1104 с.
4. Халин В.Г. Теория принятия решений. В 2 т.: учебник и практикум для бакалавриата и магистратуры/под ред. Халин В.Г., Чернова Г.В., – М.: Издательство Юрайт, 2017. – 431 С. –Серия: Бакалавр и магистр. Академический курс.
5. Ярушкина Н.Г. Основы теории нечетких и гибридных систем: учебное пособие/Н.Г. Ярушкина, – М.: Финансы и статистика, 2004. – 320 с.
6. Яськов С.Н. Введение в нейронное и когнитивное моделирование. Полный курс 2-е изд. Пер. с англ./ Яськов М.Ф., 2000
7. Байченко А.А., Байченко Л.А., Арет В.А. Применение нечеткой логики в управлении предприятием пищевой промышленности / Байченко А.А., Байченко Л.А., Арет В.А. // Научный журнал НИУ ИТМО. Серия «Экономика и экологический менеджмент» № 3, 2014
8. Владимирова Д. Б., Кокшарова А. А. Прогнозирование финансовых рынков искусственными нейронными сетями / Владимирова Д. Б., Кокшарова А. А. // Science and business: development ways
9. Гордей М.П., Смоляк Н.В. Разработка базы данных «Автозаправки» в среде MS Access / Гордей М.П., Смоляк Н.В. // Современные инновационные технологии и проблемы устойчивого развития общества. Материалы X международной научно-практической конференции. Сборник научных статей участников конференции. 2017
10. Забоев М.В., Мелешкин М.И. Оценка перспектив вхождения российских университетов в первую сотню ведущих мировых университетов с использованием нейросетевых методов кластеризации данных / Забоев М.В., Мелешкин М.И. // Прикладная информатика Vol. 10. No. 3 (57). 2015 С. 52 – 61.
11. Кузькин А.А. Оценивание показателей эффективности и результативности ИТ-процессов с использованием гибридных нейро-нечетких сетей / Кузькин А.А. // Интернет-журнал «Науковедение», 2014

12. Курбанов М.К., Ибаев Ж.Г. Общие сведения о базе данных гидрогеологических параметров артезианских скважин в приложении Microsoft Access / Курбанов М.К., Ибаев Ж.Г. // Труды Института геологии Дагестанского Научного Центра РАН
13. Новаков Р.М., Таскин В.В., Сидоров М.Д., Паламарь С.В. Разработка базы данных Шанучского месторождения (Камчатка) в среде Microsoft Access / Новаков Р.М., Таскин В.В., Сидоров М.Д., Паламарь С.В. // Вестник Камчатской региональной организации учебно-научный центр. Серия: Науки о земле
14. Поворознюк А.И., Чикина Н.А., Антонова И.В. Нечеткая экспертная система прогноза риска развития профессионально обусловленных заболеваний / Поворознюк А.И., Чикина Н.А., Антонова И.В. // Вестник Национального технического университета Харьковский политехнический институт. Серия: Информатика и моделирование № 31, 2010. С127 – 132
15. Полковникова Н.А., Курейчик В.М. Разработка модели экспертной системы на основе нечёткой логики / Полковникова Н.А., Курейчик В.М. // Известия Южного федерального университета. Технические науки № 1 (150), 2014. С 83 – 92
16. Пономаренко И.К., Надвоцкая В.В. Технологии создания базы данных в ACCESS на примере базы данных органов опеки и попечительства / Пономаренко И.К., Надвоцкая В.В. // Ползуновский альманах
17. Семененко М.Г., Лесина Т.В. Оценка эффективности инвестиционных проектов на основе формализма нечеткой логики / Семененко М.Г., Лесина Т.В. // Финансовая аналитика: проблемы и решения // № 29, 2011. С.63-68
18. Слеповичев И.И., Ирматов П.В., Комарова М.С., Бежин А.А. ОБНАРУЖЕНИЕ DDoS АТАК НЕЧЕТКОЙ НЕЙРОННОЙ СЕТЬЮ / Слеповичев И.И., Ирматов П.В., Комарова М.С., Бежин А.А. // Известия Саратовского университета. 2009.Т.9.Сер. Математика.Механика. Информатика, вып. 3
19. Феклистова И. Использование кластерного анализа при оценке эффективности стратегического управления предприятиями региона / Феклистова И. // Научный журнал № 2(7), 2016. С. 2.1-2.15
20. Amrina E., Ramadhani C., Vilsa A. L. “A Fuzzy Multi Criteria Approach for Sustainable Manufacturing Evaluation in Cement Industry”, 13th Global Conference on Sustainable Manufacturing - Decoupling Growth from Resource Use, 2016
21. Antonelli D., Stadnicka D. “Classification and efficiency estimation of mistake proofing solutions by Fuzzy Inference”, IFAC (International Federation of Automatic Control) , 2016
22. Gardner S.E Jr Exponential Smoothing: The state of the Art / Gardner S.E // Journal of Forecasting, 1985, Vol.4. P.1-28

23. Mamdani E.H., Assilian S. An Experiment in Linguistic Synthesis with Fuzzy Logic Controller / Mamdani E.H., Assilian S. // Int. J. Man-Machine Studies, 1975, Vol. 7(1). P.1-13
24. Sofiyabadia J., Kolahib B. and Valmohammadic C. Key performance indicators measurement in service business: a fuzzy VIKOR approach / Sofiyabadia J., Kolahib B. and Valmohammadic C. // Total Quality Management, 2016
25. Zadeh L. Fuzzy sets / Zadeh L. // Information and Control, 1965, Vol. 8. P. 338–353.
26. URL: <https://basegroup.ru/deductor/function/algorithm/kohonen> (дата обращения: 13.04.2018).
27. URL: <https://basegroup.ru/community/articles/som> (дата обращения: 13.04.2018).
28. URL: <https://googlesystem.blogspot.ru/2006/09/how-much-data-does-google-store.html#gsc.tab=0> (дата обращения: 14.04.2018)
29. URL: <https://www.cirrusinsight.com/blog/how-much-data-does-google-store> (дата обращения: 14.04.2018)
30. URL: https://ru.bmstu.wiki/Реляционная_база_данных (дата обращения: 14.04.2018)
31. URL: https://ru.bmstu.wiki/Реляционная_модель_данных (дата обращения: 14.04.2018)
32. URL: <https://habr.com/post/129195/> (дата обращения: 17.04.2018)
33. URL: <http://www.intuit.ru/studies/curriculums/15974/courses/87/info> (дата обращения: 17.04.2018)
34. URL: <http://www.machinelearning.ru/wiki/index.php?title=Сезонность> (дата обращения: 17.04.2018)
35. URL: <https://searchoracle.techtarget.com/definition/Very-Large-Database> (дата обращения: 18.04.2018)
36. URL: https://ru.wikipedia.org/wiki/Функция_принадлежностиПриложения (Дата обращения 08.05.2017)

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1

Название	Формула
Треугольная (trimf)	$\mu(x,a,b,c) = \begin{cases} 0 & \text{если } x \leq a, \\ \frac{x-a}{b-a} & \text{если } a \leq x \leq b, \\ \frac{c-x}{c-b} & \text{если } b \leq x \leq c, \\ 0 & \text{если } x \geq c. \end{cases}$
Трапецевидная (trapmf)	$\mu(x;a,b,c,d) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$
Колоколообразная (gbellmf)	$\mu(x;a,b,c) = \frac{1}{1 + \left \frac{x-c}{a} \right ^{2b}}$
Гауссова (gaussmf)	$\mu(x;a,c) = \frac{1}{1 + e^{-a(x-c)}}$
Сигмоидальная (sigmf)	$\mu(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}, \quad \sigma > 0$

Таблица 2.

Наименование операции	Содержание операции
Включение	Нечеткое множество A содержится в B , если: $\mu_A(x) \leq \mu_B(x), \forall x \in X$
Равенство	Множества A и B равны, если: $\mu_A(x) = \mu_B(x), \forall x \in X$
Дополнение	Множество B является дополнительным A ,

	<p>если: $\mu_B(x) = 1 - \mu_A(x), \forall x \in X$,</p> <p>при условии, что $0 \leq \mu_A(x) \leq 1$, что не умоляет общности</p>
Пересечение	<p>Пересечением нечетких множеств A и B называется нечеткое множество $A \cap B$ с функцией принадлежности:</p> $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), \forall x \in X$
Объединение	<p>Объединением нечетких множеств A и B называется нечеткое множество $A \cup B$ с функцией принадлежности:</p> $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), \forall x \in X$

Таблица 3.

ПРИЛОЖЕНИЕ 2

Информационный объект	Таблица	Атрибуты	Поля таблицы	Типы данных
Достигнутые значения по контрольному элементу ежемесячно (поквартально)	AchievementMonitorControlItem	Код контрольного элемента	IDcontrolitem	Числовой
		Код месяца	IDmonth	Числовой
		Фактический результат	actualresult	Текстовый
		Комментарий	comment	Текстовый
		Финансовый год	fy	Числовой
		Скорректированное значение контр. эл.	targetLatestMonth	Текстовый
		Прогноз на конец года	EOYforecast	Текстовый
Достигнутые значения по таргету	AchievementMonitorTarget	Код таргета	IDtarget	Числовой
		Код месяца	IDmonth	Числовой
		Фактический	actualresult	Текстовый

помесячно (поквартально)		результат		
		Метод вычисления итога	method	Текстовый
		Комментарий	comment	Текстовый
		Финансовый год	fy	Числовой
		Скорректирован ное значение таргета	targetLatestMonth	Текстовый
		Прогноз на конец года	EOYforecast	Текстовый
Категории	Category	Код категории	IDcategory	Счетчик
		Название категории	category	Текстовый
		Видимость для заполняющего	isvisible	Логический
Контрольные элементы	CI	Код контрольного элемента	IDcontrolitem	Счетчик
		Контрольный элемент	controlitem	Текстовый
		Год внесения	yearcontrolitem	Числовой
		Целевое значение	targetvalue	Денежный
		Обязательный минимум	commitmentvalue	Денежный
		Код периодичности	IDtiming	Числовой
		Код метода подсчета результата	IDmethod	Числовой
		Код ЕИ	IDUI	Текстовый
Ответственные	CIResponsible	Код	IDcontrolitem	Числовой

		контрольного элемента		
		Код ответственного	IDresponsible	Счетчик
		Код сотрудника	IDstuffmember	Числовой
Поддерживаю щие	CISupporter	Код контрольного элемента	IDcontrolitem	Числовой
		Кот поддерживающе го	IDSupporter	Счетчик
		Код сотрудника	IDstuffmember	Числовой
Цель- Характеристика	Columns	Код характеристики цели	IDcn	Числовой
		Код цели	IDpurpose	Числовой
Характеристик и цели	ColumnsNames	Код характеристики цели	IDcn	Счетчик
		Название характеристики	columnname	Текстовый
		Видимость для заполняющего	isvisible	Логический
Методы подсчета результата	Methods	Код метода подсчета результата	IDmethod	Счетчик
		Название метода	method	Текстовый
Месяцы	Months	Код месяца	IDmonth	Счетчик
		Название месяца	namemonth	Текстовый
Цели	Purpose	Код цели	IDpurpose	Счетчик
		Цель	purpose	Текстовый
		Код категории	IDcategory	Числовой
		Код сотрудника	IDstuffmember	Числовой
Кварталы	Quarters	Код месяца	IDmonth	Числовой

		Месяц, последний в квартале	monthquarter	Текстовый
Стратегии	Strategy	Код стратегии	IDstrategy	Счетчик
		Стратегия	strategy	Текстовый
		Код цели	IDpurpose	Числовой
Стратегия- Контрольный элемент	StrategyControlItem	Код стратегии	IDstrategy	Числовой
		Код контрольного элемента	IDcontrolitem	Числовой
Сотрудники	StuffMember	Код сотрудника	IDstuffmember	Счетчик
		ФИО сотрудника	namestuffmem ber	Текстовый
		Должность	EVPllevel	Текстовый
		Отдел	department	Текстовый
		e-mail	mail	Текстовый
Таргеты	Target	Код таргета	IDtarget	Счетчик
		Таргет	Target	Текстовый
		Код периодичности	IDtiming	Числовой
		Код цели	IDpurpose	Числовой
		Код метода	IDmethod	Числовой
		Код ЕИ	IDUI	Текстовый
		Уровень ответственности за выполнение	respsupp	Текстовый
		Код делегирувавшего	IDdeligator	Числовой
		Время делегирувания	TimeDeligatio n	Дата/время
Значения таргетов по годам	TargetYears	Код таргета	IDtarget	Числовой
		Год внесения	yeartarget	Числовой
		Целевое значение	targetvalue	Денежный

		Обязательный минимум	commitmentvalue	Денежный
Периодичность измерения	Timing	Код периодичности	IDtiming	Счетчик
		Периодичность измерения	timing	Текстовый
Единицы измерения	UI	Код ЕИ	IDUI	Счетчик
		ЕИ	UI	Текстовый

Таблица 4.

ПРИЛОЖЕНИЕ 3

Choose stuff member

Zholudeva Ksenia

Choose

Add Cathegory to the Handbook

Add Objective characteristics to the Handbook

Add stuff member

Рис. 28. Выбор сотрудника через форму.

Zholudeva Ksenia

Back

AOP

Open

Edit

RF1

Open

Edit

Рис. 29. Выбор внесения либо просмотра: фактические результаты или плананове целевые значения.

Zholudeva Ksenia AOP: Objectives Back GLOBAL OBJECTIVE:

Objective	GK for PC section managers (NPCW training is included)	Characteristics	Targets	Strategies	Talent Pool Reinforcement
Category	People				
Objective	Develop action plan to optimize inland stock (EWH + Plant) to satisfy customer: safety stock management, RLQ/SNP	Characteristics	Targets	Strategies	Reduce in-Plant and Total Parts Inventory
Category	MTP 2020				
Objective	Reduce mislabeling	Characteristics	Targets	Strategies	
Category	Cost				

Рис. 30. Внесение стратегических целей.

Characteristics of the objective: Reduce mislabeling Back

R&D Costs

Рис. 31. Выбор характеристик цели из редактируемого справочника.

Targets of the objective: Reduce mislabeling Back

Target: Number of parts with mislabeling UI %

Timing: monthly

Method: EOY

Look at other your targets

Year target	Target Value	commitmentvalue
2018	0,00	10,00
*	0,00	0,00

Deligated by TimeDeligation You are

Запись: 1 из 1 Нет фильтра Поиск

Рис. 32. Внесение таргета для цели.

Strategies for objective: **Reduce mislabeling** Back

Choose Target: **Number of mislabeling parts, that are not in the work**

Strategy: **Plan the time effectively** Look at other your strategies

Controlitem	spend on 1 vendor minimum time daily	UI	minutes	Year	2017	Responsible
Timing	monthly			Target value	15,00	Supporters
Method	EOY			Commitment	30,00	
Controlitem		UI		Year		Responsible
Timing				Target value		Supporters
Method				Commitment		

Рис. 33. Внесение стратегий цели, а также контрольных элементов.

Responsible for control item: **spend on 1 vendor minimum time daily** Back

Responsible: **Afanasenko Irina**

Responsible: **Alekseeva Tatiana**

Рис. 34. Назначение ответственных за контрольный элемент стратегии.

Zholudeva Ksenia **RF1** Back

What values do you want to add?

Target

Control item

Рис. 35. Выбор варианта внесения фактического результата: таргет или контрольный элемент.

Targets of stuffmember: Zholudeva Ksenia Back

Target	Number of vendors explanatory letters lacking	
Target	Number of parts with mislabeling	
Target	Number of mislabeling parts, that are not in the work	
Target		

Microsoft Access

Write a financial year, you are interested in, or agree with suggested value

2018

OK Cancel

Рис. 36. Выбор таргета цели в разделе занесения фактических результатов, а также выбор интересующего финансового года (по умолчанию выводится текущий год).

RF1: Objective: Target: Number of parts with mislabeling Values Financial Year 2018 Back

Month	april	Value	0.6	Comment	Vendor's packaging machine temporarily is out of order
Month		Value		Comment	

Рис. 37. Внесение фактического результата и комментария.

ПРИЛОЖЕНИЕ 4

Описание работы блока WRONG MARKING

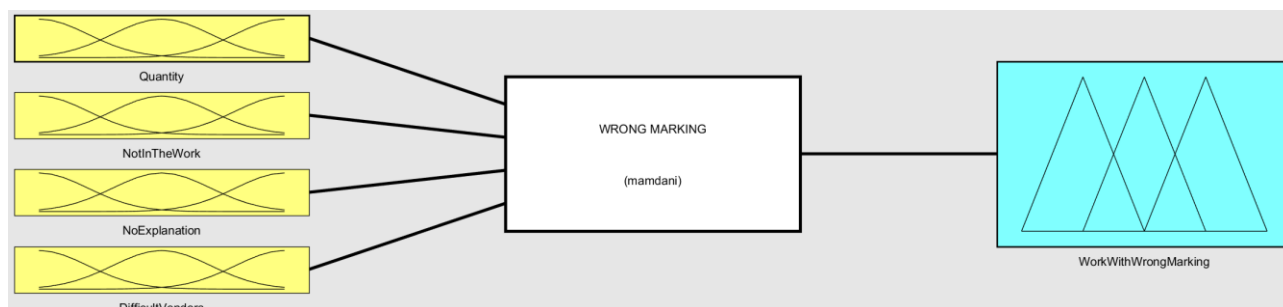


Рис. 38. Блок WRONG MARKING

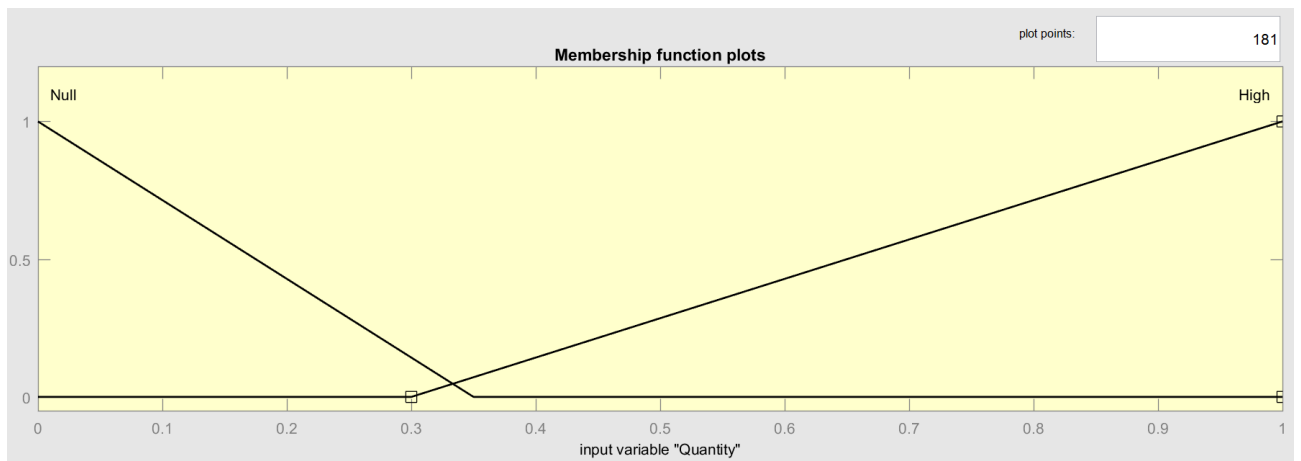


Рис. 39. Лингвистическая переменная входа Quantity, блок WRONG MARKING

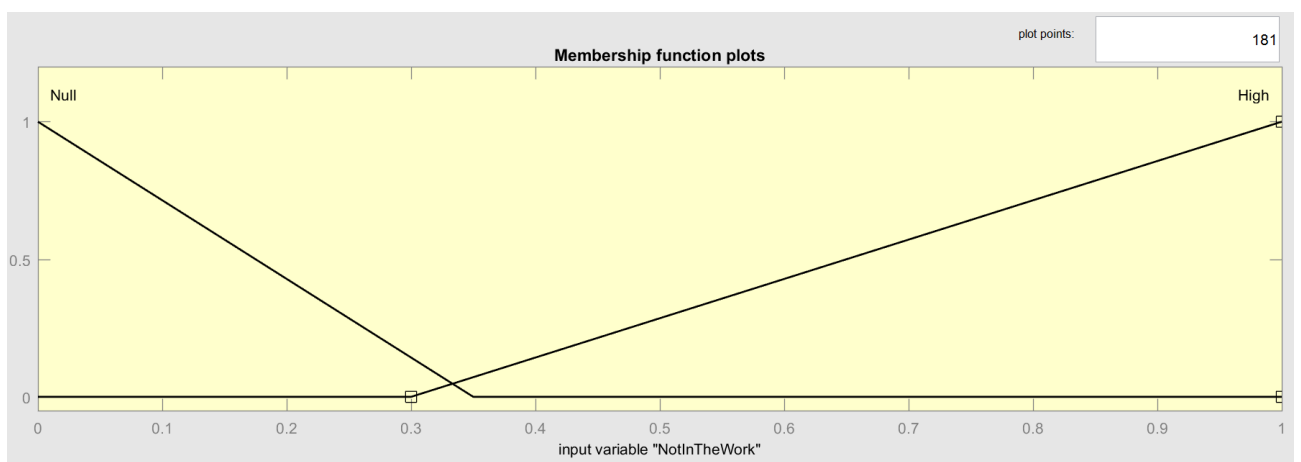


Рис. 40. Лингвистическая переменная входа NotInTheWork, блок WRONG MARKING

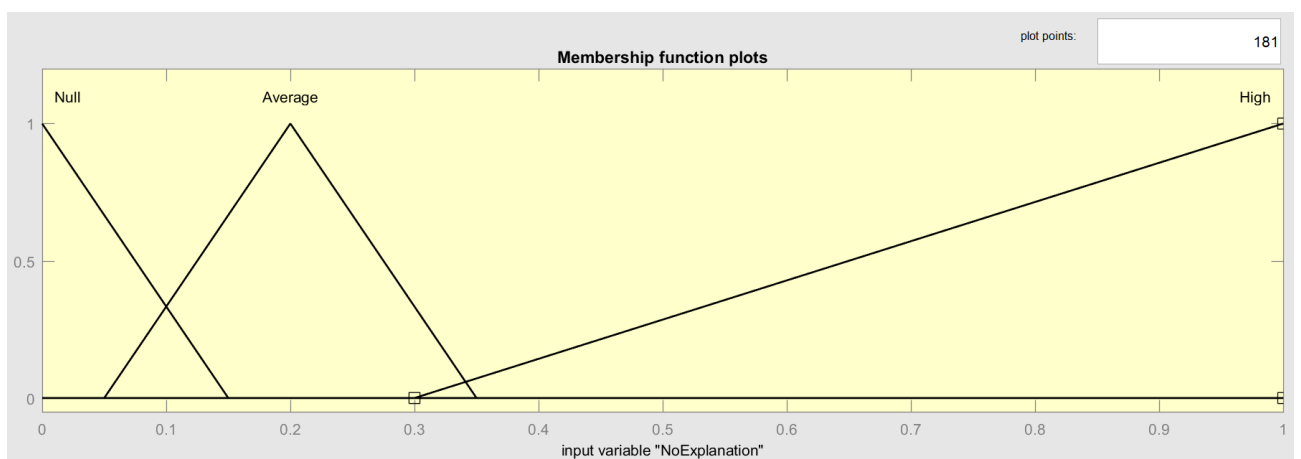


Рис. 41. Лингвистическая переменная входа NoExplanation, блок WRONG MARKING

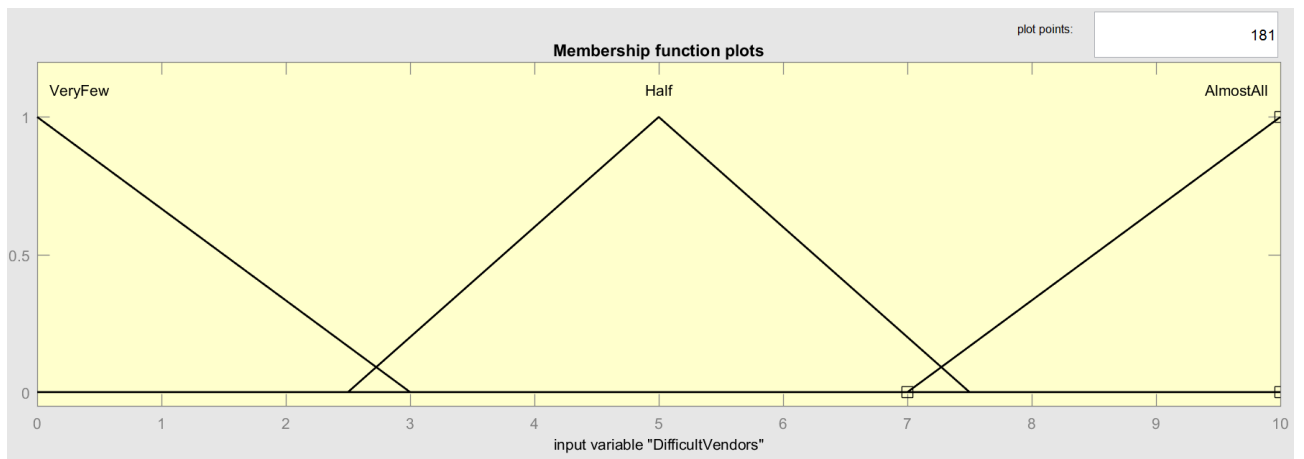


Рис. 42. Лингвистическая переменная входа DifficultVendors, блок WRONG MARKING

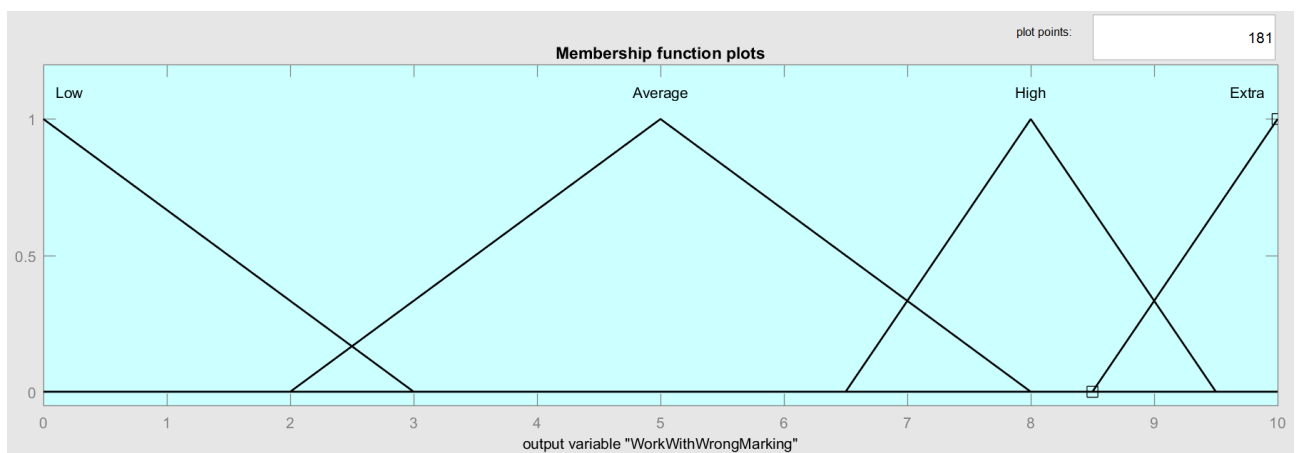


Рис. 43. Лингвистическая переменная выхода WorkWithWrongMarking, блок WRONG MARKING

1. If (Quantity is Null) and (NotInTheWork is Null) and (NoExplanation is Null) and (DifficultVendors is VeryFew) then (WorkWithWrongMarking is Average) (1)
2. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is Average) and (DifficultVendors is VeryFew) then (WorkWithWrongMarking is Average) (1)
3. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is High) and (DifficultVendors is VeryFew) then (WorkWithWrongMarking is Average) (1)
4. If (NotInTheWork is High) and (DifficultVendors is VeryFew) then (WorkWithWrongMarking is Low) (1)
5. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is Null) and (DifficultVendors is VeryFew) then (WorkWithWrongMarking is High) (1)
6. If (Quantity is Null) and (NotInTheWork is Null) and (NoExplanation is Null) and (DifficultVendors is Half) then (WorkWithWrongMarking is Average) (1)
7. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is Average) and (DifficultVendors is Half) then (WorkWithWrongMarking is High) (1)
8. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is High) and (DifficultVendors is Half) then (WorkWithWrongMarking is Average) (1)
9. If (NotInTheWork is High) and (DifficultVendors is Half) then (WorkWithWrongMarking is Low) (1)
10. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is Null) and (DifficultVendors is Half) then (WorkWithWrongMarking is High) (1)
11. If (Quantity is Null) and (NotInTheWork is Null) and (NoExplanation is Null) and (DifficultVendors is AlmostAll) then (WorkWithWrongMarking is Average) (1)
12. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is Average) and (DifficultVendors is AlmostAll) then (WorkWithWrongMarking is High) (1)
13. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is High) and (DifficultVendors is AlmostAll) then (WorkWithWrongMarking is Average) (1)
14. If (NotInTheWork is High) and (DifficultVendors is AlmostAll) then (WorkWithWrongMarking is Low) (1)
15. If (Quantity is High) and (NotInTheWork is Null) and (NoExplanation is Null) and (DifficultVendors is AlmostAll) then (WorkWithWrongMarking is Extra) (1)
16. If (Quantity is Null) and (NotInTheWork is Null) and (NoExplanation is Average) and (DifficultVendors is VeryFew) then (WorkWithWrongMarking is Low) (1)
17. If (Quantity is Null) and (NotInTheWork is Null) and (NoExplanation is Average) and (DifficultVendors is Half) then (WorkWithWrongMarking is Average) (1)

Рис. 44. База правил блока WRONG MARKING, 1-17 правила из 18

18. If (Quantity is Null) and (NotInTheWork is Null) and (NoExplanation is Average) and (DifficultVendors is AlmostAll) then (WorkWithWrongMarking is Average) (1)

Рис. 45. База правил блока WRONG MARKING, 18 правило из 18

Описание работы блока RISK PARTS

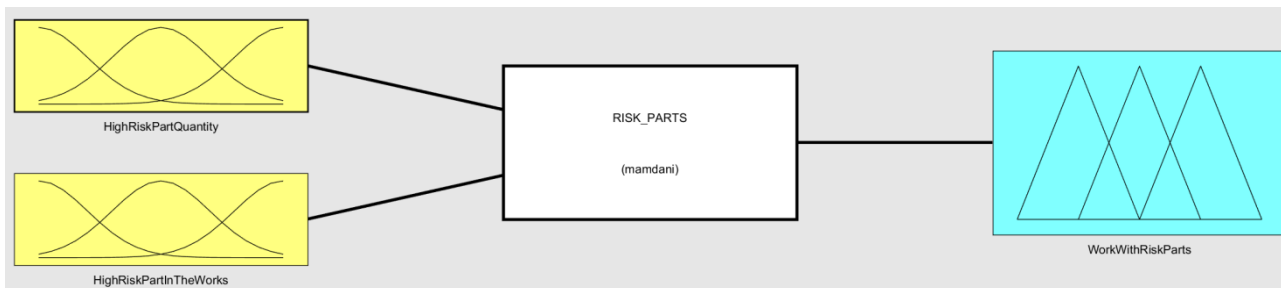


Рис. 46. Блок RISK PARTS

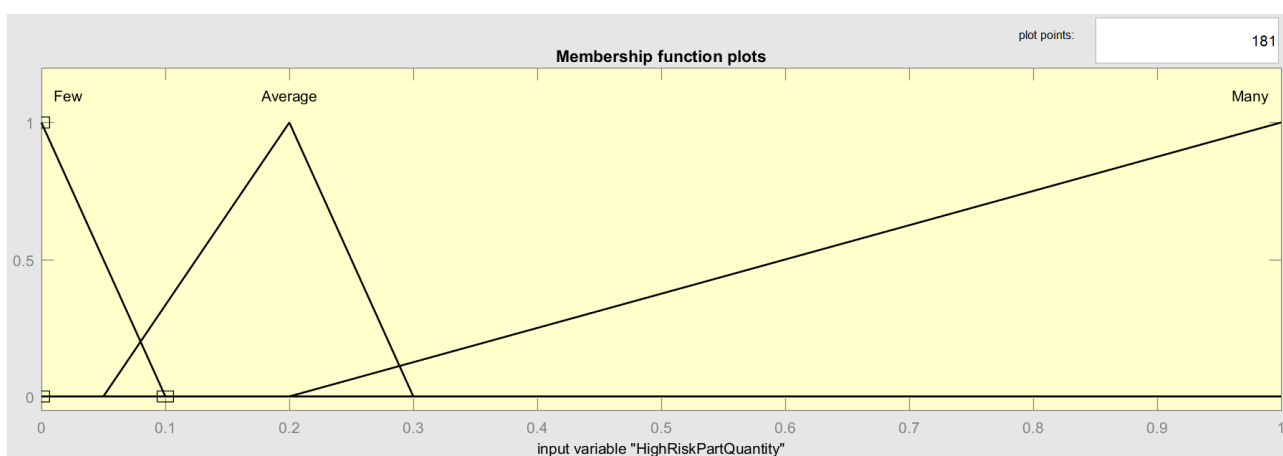


Рис. 47. Лингвистическая переменная входа HighRiskPartQuantity, блок RISK PARTS

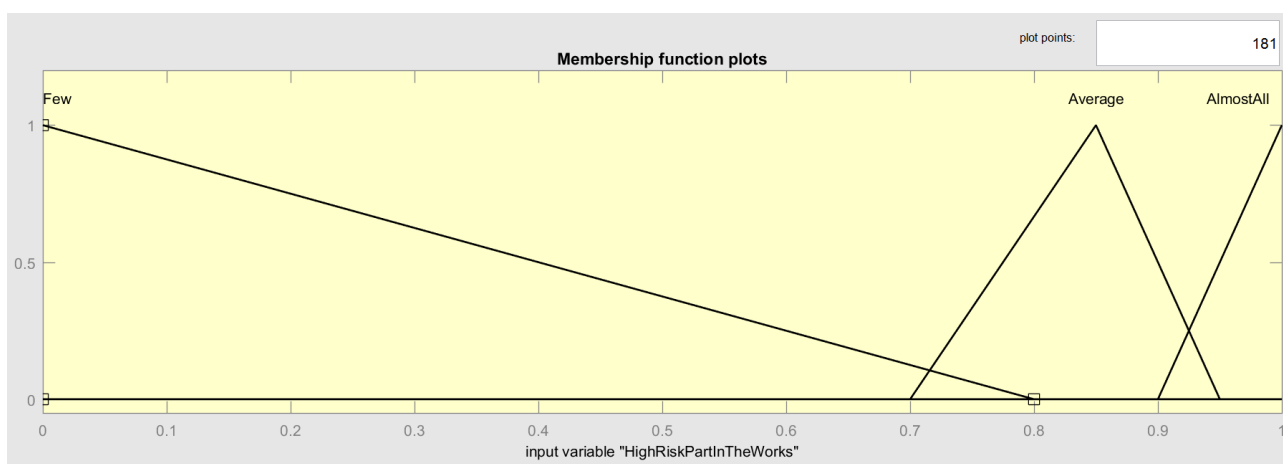


Рис. 48. Лингвистическая переменная входа HighRiskPartInTheWorks, блок RISK PARTS

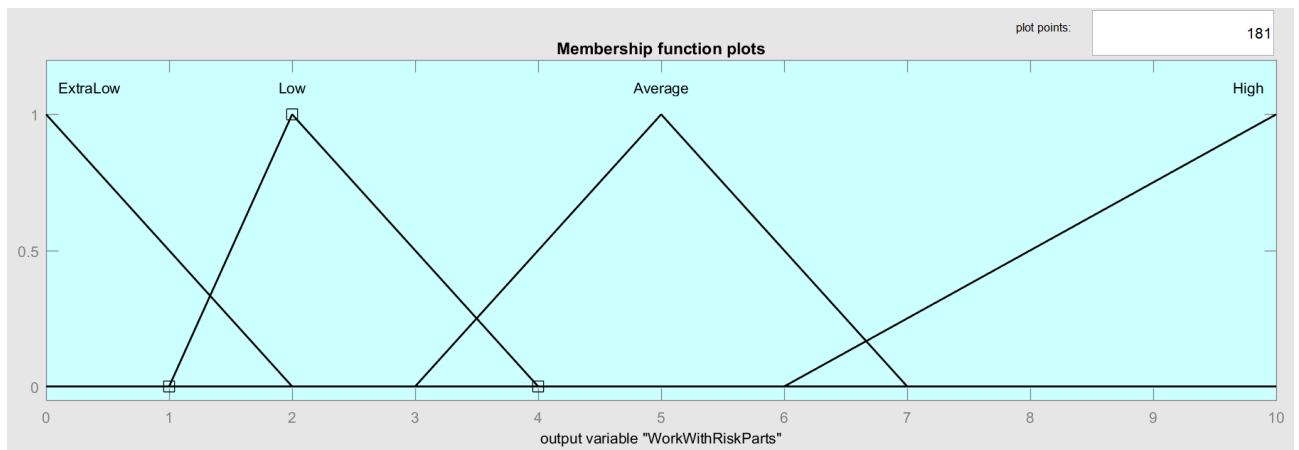


Рис. 49. Лингвистическая переменная выхода WorkWithRiskParts, блок RISK PARTS

1. If (HighRiskPartQuantity is Few) and (HighRiskPartInTheWorks is AlmostAll) then (WorkWithRiskParts is High) (1)
2. If (HighRiskPartQuantity is Average) and (HighRiskPartInTheWorks is AlmostAll) then (WorkWithRiskParts is High) (1)
3. If (HighRiskPartQuantity is Many) and (HighRiskPartInTheWorks is AlmostAll) then (WorkWithRiskParts is High) (1)
4. If (HighRiskPartQuantity is Few) and (HighRiskPartInTheWorks is Average) then (WorkWithRiskParts is ExtraLow) (1)
5. If (HighRiskPartQuantity is Average) and (HighRiskPartInTheWorks is Average) then (WorkWithRiskParts is Low) (1)
6. If (HighRiskPartQuantity is Many) and (HighRiskPartInTheWorks is Average) then (WorkWithRiskParts is Average) (1)
7. If (HighRiskPartQuantity is Few) and (HighRiskPartInTheWorks is Few) then (WorkWithRiskParts is ExtraLow) (1)
8. If (HighRiskPartQuantity is Average) and (HighRiskPartInTheWorks is Few) then (WorkWithRiskParts is Low) (1)
9. If (HighRiskPartQuantity is Many) and (HighRiskPartInTheWorks is Few) then (WorkWithRiskParts is Low) (1)

Рис. 50. База правил блока RISK PARTS, 9 правил

Описание работы блока SAFETY DAYS

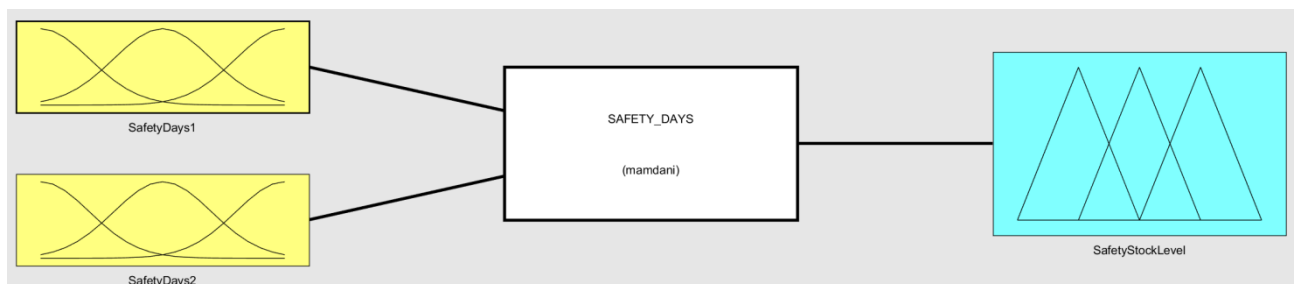


Рис. 51. Блок SAFETY DAYS

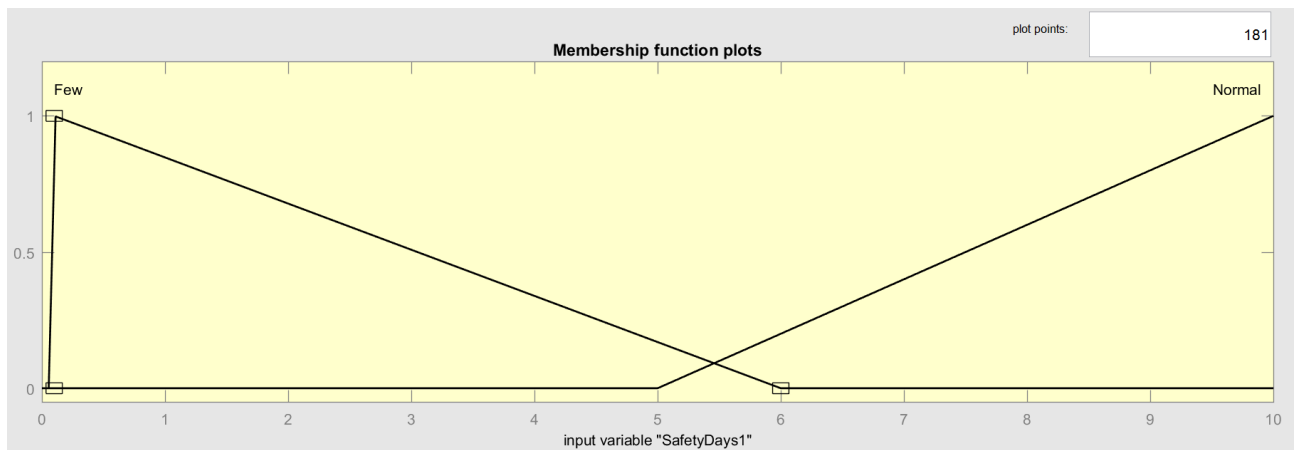


Рис. 52. Лингвистическая переменная входа SafetyDays1, блок SAFETY DAYS

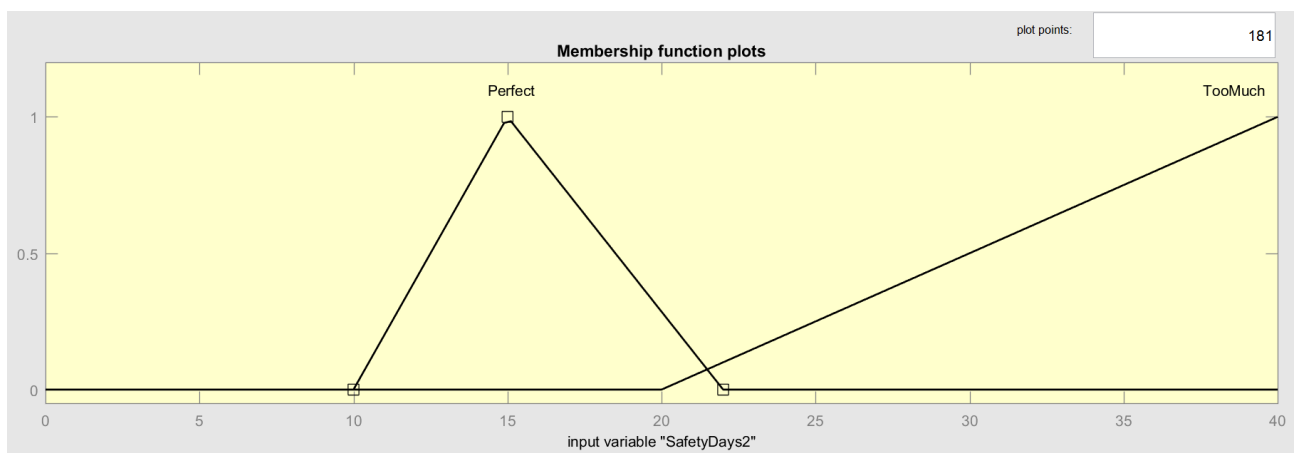


Рис. 53. Лингвистическая переменная входа SafetyDays2, блок SAFETY DAYS

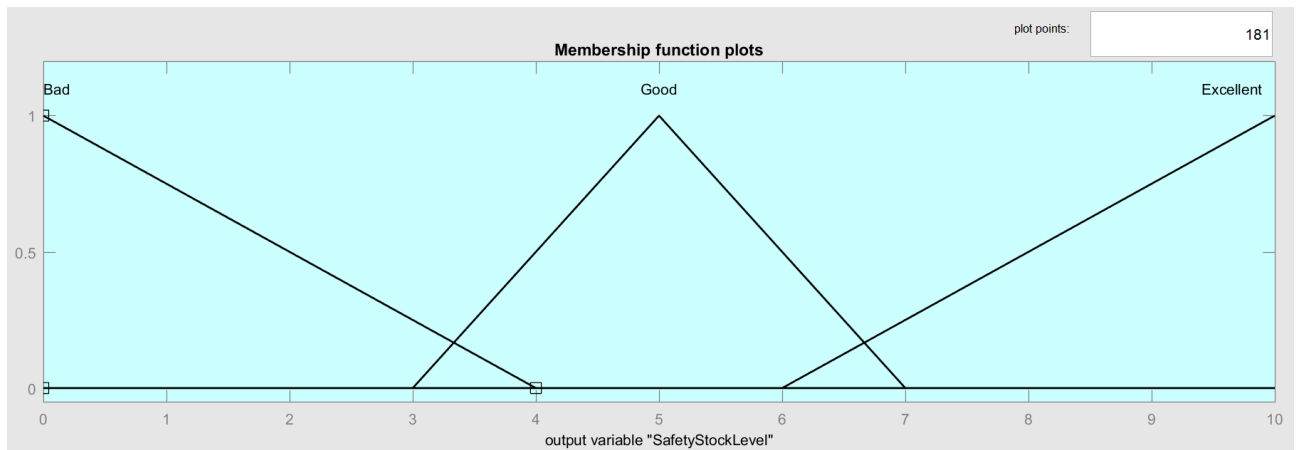


Рис. 54. Лингвистическая переменная выхода SafetyStockLevel, блок SAFETY DAYS

1. If (SafetyDays1 is Few) then (SafetyStockLevel is Bad) (1)
2. If (SafetyDays1 is Normal) then (SafetyStockLevel is Good) (1)
3. If (SafetyDays2 is Perfect) then (SafetyStockLevel is Excellent) (1)
4. If (SafetyDays2 is TooMuch) then (SafetyStockLevel is Bad) (1)
5. If (SafetyDays1 is Few) and (SafetyDays2 is Perfect) then (SafetyStockLevel is Bad) (1)
6. If (SafetyDays1 is Normal) and (SafetyDays2 is Perfect) then (SafetyStockLevel is Good) (1)
7. If (SafetyDays1 is Few) and (SafetyDays2 is TooMuch) then (SafetyStockLevel is Bad) (1)
8. If (SafetyDays1 is Normal) and (SafetyDays2 is TooMuch) then (SafetyStockLevel is Good) (1)

Рис. 55. База правил блока SAFETY DAYS, 8 правил

Описание работы блока SELF-DEVELOPMENT

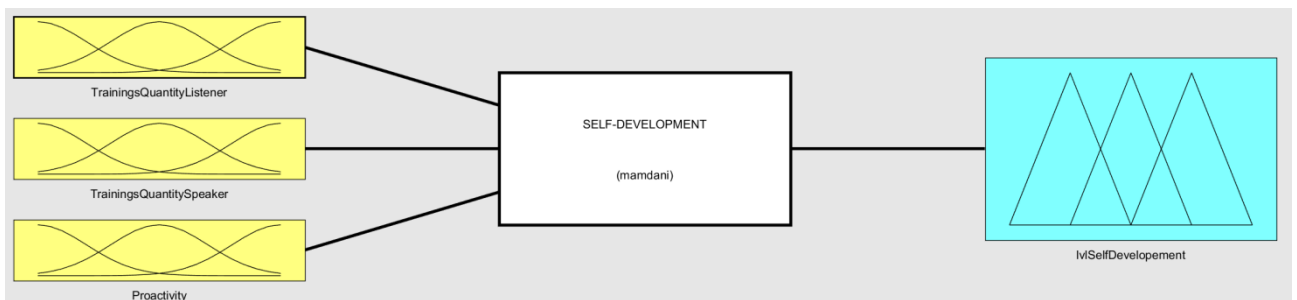


Рис. 56. Блок SELF-DEVELOPMENT



Рис. 57. Лингвистическая переменная входа TrainingsQuantityListener, блок SELF-DEVELOPMENT

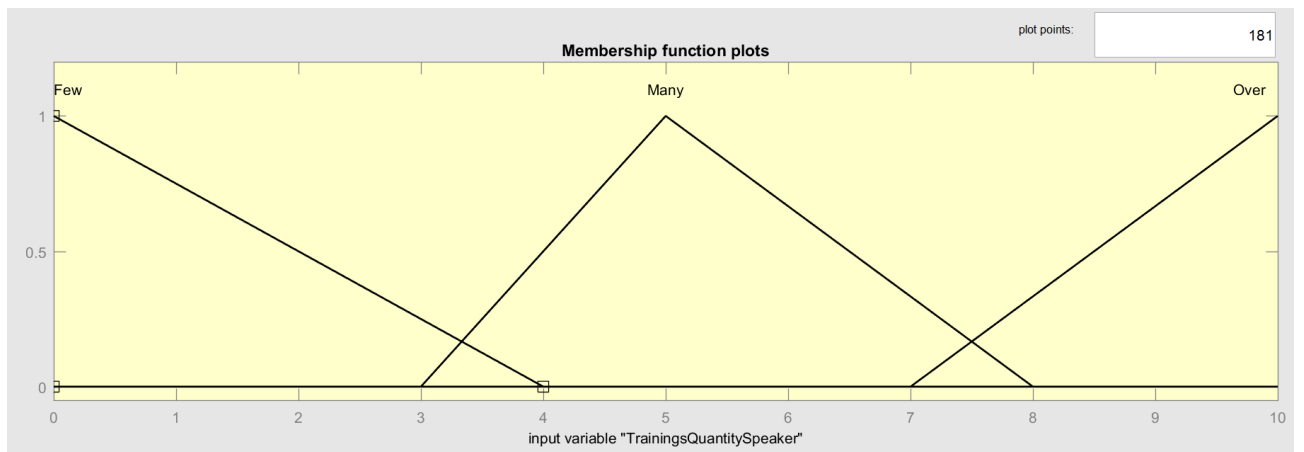


Рис. 58. Лингвистическая переменная входа TrainingsQuantitySpeaker, блок SELF-DEVELOPMENT

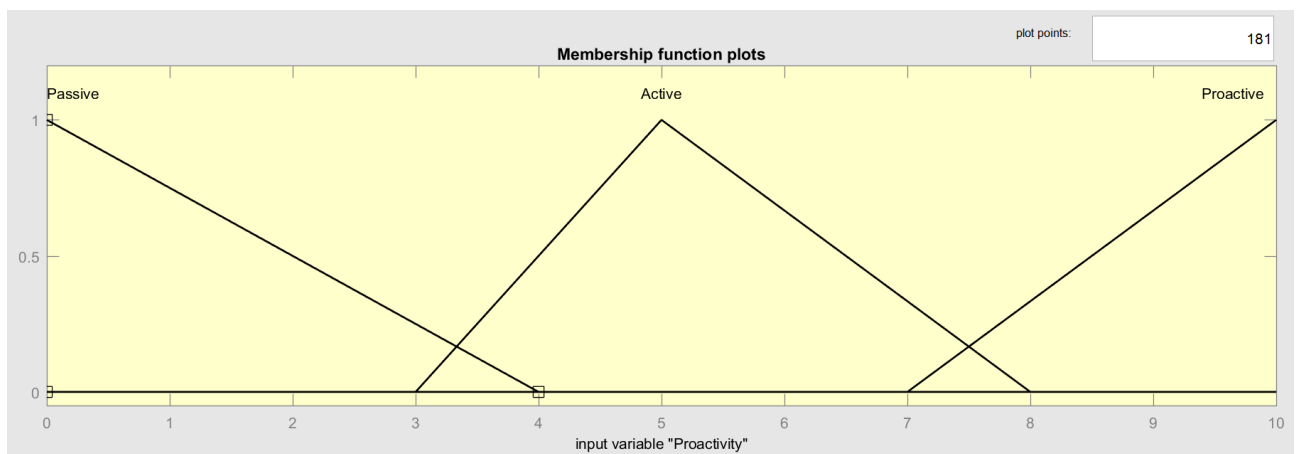


Рис. 59. Лингвистическая переменная входа Proactivity, блок SELF-DEVELOPMENT

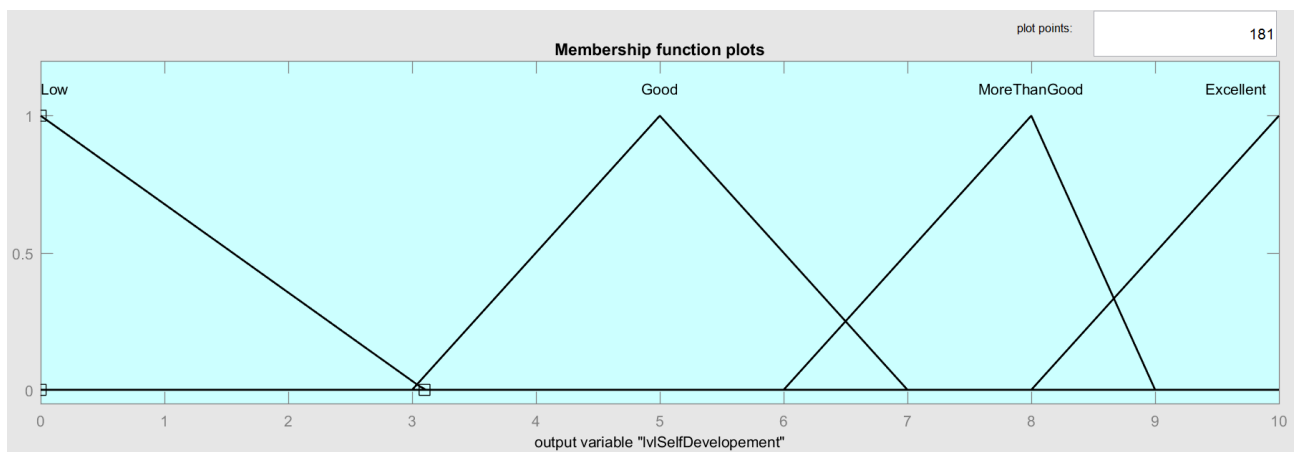


Рис. 60. Лингвистическая переменная выхода LevelSelfDevelopment, блок SELF-DEVELOPMENT

1. If (TrainingsQuantityListener is Few) and (TrainingsQuantitySpeaker is Few) and (Proactivity is Passive) then (LevelSelfDevelopment is Low) (1)
2. If (TrainingsQuantityListener is Many) and (TrainingsQuantitySpeaker is Many) and (Proactivity is Active) then (LevelSelfDevelopment is MoreThanGood) (1)
3. If (TrainingsQuantityListener is Over) and (TrainingsQuantitySpeaker is Over) and (Proactivity is Proactive) then (LevelSelfDevelopment is Excellent) (1)
4. If (TrainingsQuantityListener is Few) and (TrainingsQuantitySpeaker is Many) and (Proactivity is Passive) then (LevelSelfDevelopment is Good) (1)
5. If (TrainingsQuantityListener is Few) and (TrainingsQuantitySpeaker is Many) and (Proactivity is Active) then (LevelSelfDevelopment is MoreThanGood) (1)
6. If (TrainingsQuantityListener is Few) and (TrainingsQuantitySpeaker is Many) and (Proactivity is Proactive) then (LevelSelfDevelopment is MoreThanGood) (1)
7. If (TrainingsQuantityListener is Few) and (TrainingsQuantitySpeaker is Over) and (Proactivity is Active) then (LevelSelfDevelopment is MoreThanGood) (1)
8. If (TrainingsQuantityListener is Few) and (TrainingsQuantitySpeaker is Over) and (Proactivity is Proactive) then (LevelSelfDevelopment is MoreThanGood) (1)
9. If (TrainingsQuantityListener is Few) and (TrainingsQuantitySpeaker is Over) and (Proactivity is Proactive) then (LevelSelfDevelopment is Excellent) (1)
10. If (TrainingsQuantityListener is Many) and (TrainingsQuantitySpeaker is Few) and (Proactivity is Active) then (LevelSelfDevelopment is MoreThanGood) (1)
11. If (TrainingsQuantityListener is Many) and (TrainingsQuantitySpeaker is Few) and (Proactivity is Proactive) then (LevelSelfDevelopment is MoreThanGood) (1)
12. If (TrainingsQuantityListener is Many) and (TrainingsQuantitySpeaker is Many) and (Proactivity is Proactive) then (LevelSelfDevelopment is Excellent) (1)
13. If (TrainingsQuantityListener is Many) and (TrainingsQuantitySpeaker is Over) and (Proactivity is Proactive) then (LevelSelfDevelopment is Excellent) (1)
14. If (TrainingsQuantityListener is Over) and (TrainingsQuantitySpeaker is Few) and (Proactivity is Active) then (LevelSelfDevelopment is MoreThanGood) (1)
15. If (TrainingsQuantityListener is Over) and (TrainingsQuantitySpeaker is Many) and (Proactivity is Active) then (LevelSelfDevelopment is Excellent) (1)
16. If (TrainingsQuantityListener is Over) and (TrainingsQuantitySpeaker is Over) and (Proactivity is Active) then (LevelSelfDevelopment is Excellent) (1)
17. If (TrainingsQuantityListener is Many) and (TrainingsQuantitySpeaker is Over) and (Proactivity is Active) then (LevelSelfDevelopment is MoreThanGood) (1)

Рис. 61. База правил блока SELF-DEVELOPMENT, 17 правил

Описание работы блока WORK PERFECTION

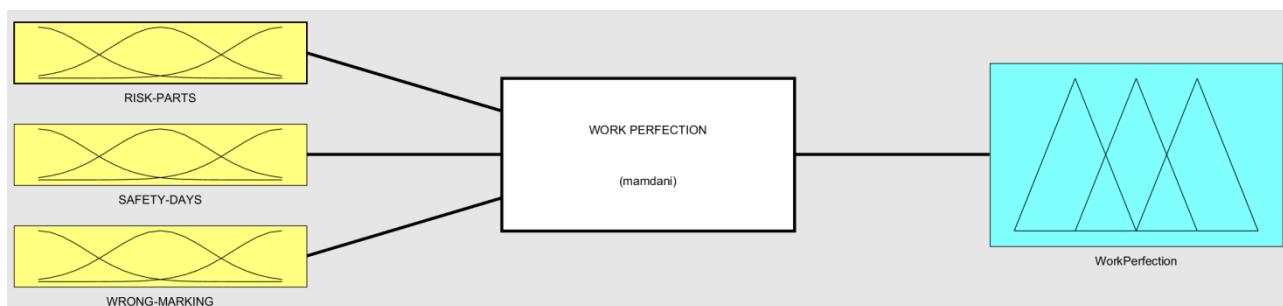


Рис. 62. Блок WORK PERFECTION

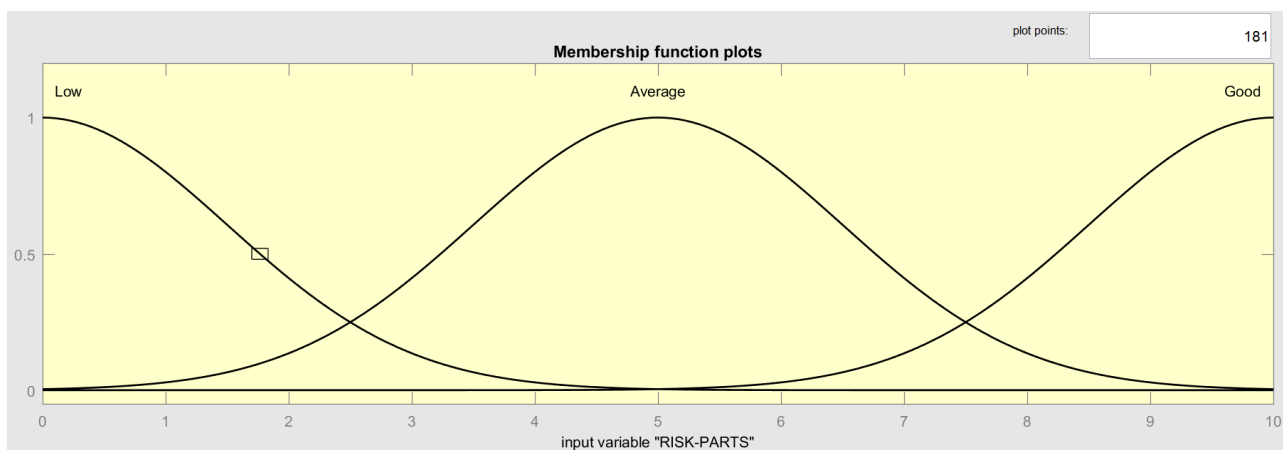


Рис. 63. Лингвистическая переменная входа RISK-PARTS, блок WORK PERFECTION

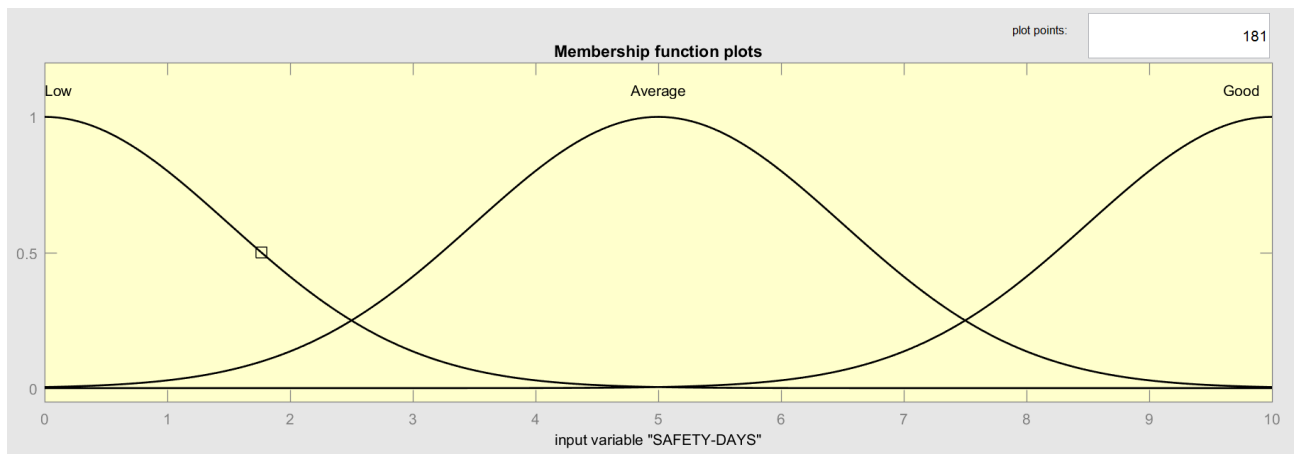


Рис. 64. Лингвистическая переменная входа SAFETY-DAYS, блок WORK PERFECTION

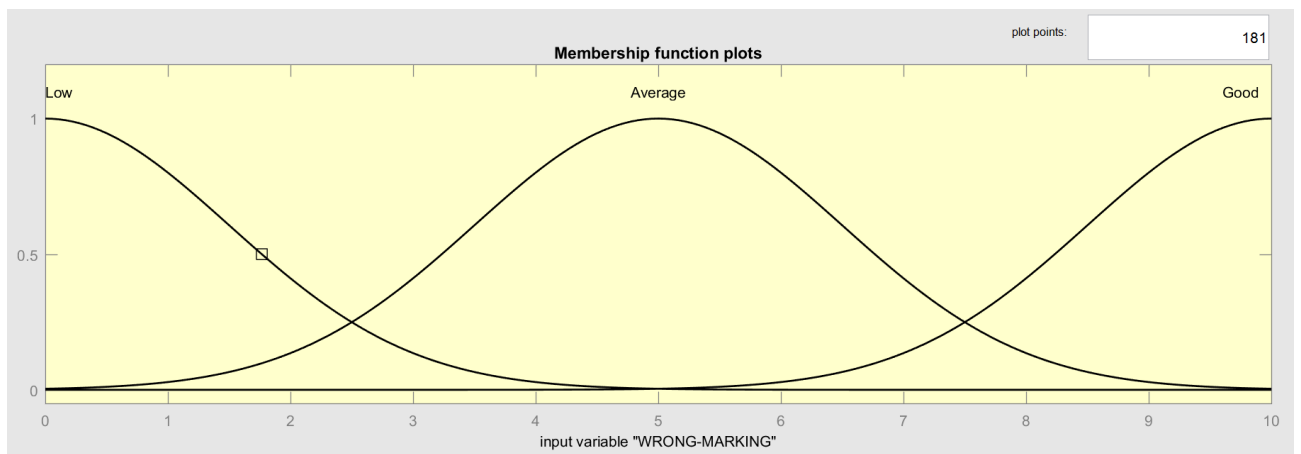


Рис. 65. Лингвистическая переменная входа WRONG-MARKING, блок WORK PERFECTION

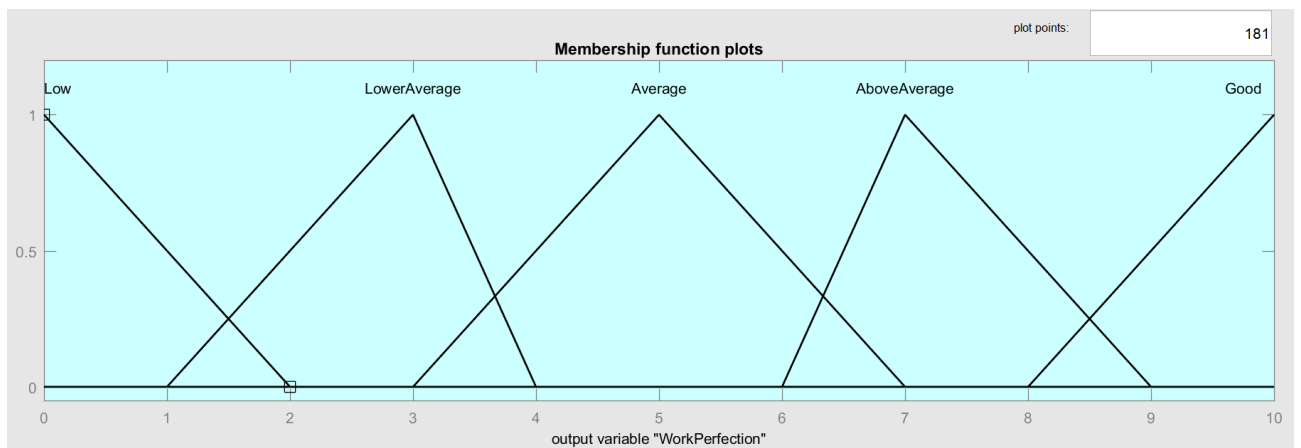


Рис. 66. Лингвистическая переменная выхода WorkPerfection, блок WORK PERFECTION


```

1. If (RISK-PARTS is Low) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Low) then (WorkPerfection is Low) (1)
2. If (RISK-PARTS is Average) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Average) then (WorkPerfection is Average) (1)
3. If (RISK-PARTS is Good) and (SAFETY-DAYS is Good) and (WRONG-MARKING is Good) then (WorkPerfection is Good) (1)
4. If (RISK-PARTS is Low) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Average) then (WorkPerfection is LowerAverage) (1)
5. If (RISK-PARTS is Low) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Low) then (WorkPerfection is LowerAverage) (1)
6. If (RISK-PARTS is Average) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Low) then (WorkPerfection is LowerAverage) (1)
7. If (RISK-PARTS is Low) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Good) then (WorkPerfection is LowerAverage) (1)
8. If (RISK-PARTS is Low) and (SAFETY-DAYS is Good) and (WRONG-MARKING is Low) then (WorkPerfection is LowerAverage) (1)
9. If (RISK-PARTS is Good) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Low) then (WorkPerfection is LowerAverage) (1)
10. If (RISK-PARTS is Low) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Average) then (WorkPerfection is Average) (1)
11. If (RISK-PARTS is Average) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Low) then (WorkPerfection is Average) (1)
12. If (RISK-PARTS is Average) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Average) then (WorkPerfection is Average) (1)
13. If (RISK-PARTS is Low) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Good) then (WorkPerfection is Average) (1)
14. If (RISK-PARTS is Good) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Low) then (WorkPerfection is Average) (1)
15. If (RISK-PARTS is Low) and (SAFETY-DAYS is Good) and (WRONG-MARKING is Average) then (WorkPerfection is Average) (1)
16. If (RISK-PARTS is Average) and (SAFETY-DAYS is Good) and (WRONG-MARKING is Low) then (WorkPerfection is Average) (1)
17. If (RISK-PARTS is Good) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Average) then (WorkPerfection is Average) (1)

```

Рис. 67. База правил блока WORK PERFECTION, 1-17 правила из 24

```

18. If (RISK-PARTS is Average) and (SAFETY-DAYS is Low) and (WRONG-MARKING is Good) then (WorkPerfection is Average) (1)
19. If (RISK-PARTS is Average) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Good) then (WorkPerfection is AboveAverage) (1)
20. If (RISK-PARTS is Average) and (SAFETY-DAYS is Good) and (WRONG-MARKING is Average) then (WorkPerfection is AboveAverage) (1)
21. If (RISK-PARTS is Good) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Average) then (WorkPerfection is AboveAverage) (1)
22. If (RISK-PARTS is Good) and (SAFETY-DAYS is Good) and (WRONG-MARKING is Average) then (WorkPerfection is Good) (1)
23. If (RISK-PARTS is Good) and (SAFETY-DAYS is Average) and (WRONG-MARKING is Good) then (WorkPerfection is Good) (1)
24. If (RISK-PARTS is Average) and (SAFETY-DAYS is Good) and (WRONG-MARKING is Good) then (WorkPerfection is Good) (1)

```

Рис. 68. База правил блока WORK PERFECTION, 18-24 правила из 24

ПРИЛОЖЕНИЕ 5

Пример кода MATLAB для реализации работы экспертной системы с заданными входами in, где экспертная система записана в модуль Sim.slx; результат записывается в переменную out:

```
in=[0.5 0 0.01 6 0.1 1 9 15 13 8 10];
```

```
i=1; %обрабатываем единственную строку-пример
```

```

simIn = Simulink.SimulationInput('Sim');
simIn = simIn.setBlockParameter('Sim/Quantity','Value', 'in(i,1)');
simIn = simIn.setBlockParameter('Sim/NotInTheWork','Value', 'in(i,2)');
simIn = simIn.setBlockParameter('Sim/NoExplanation','Value', 'in(i,3)');
simIn = simIn.setBlockParameter('Sim/DifficultVendors','Value', 'in(i,4)');
simIn = simIn.setBlockParameter('Sim/HighRiskPartQuantity','Value', 'in(i,5)');
simIn = simIn.setBlockParameter('Sim/HighRiskPartInTheWork','Value', 'in(i,6)');
simIn = simIn.setBlockParameter('Sim/SafetyDays1','Value', 'in(i,7)');
simIn = simIn.setBlockParameter('Sim/SafetyDays2','Value', 'in(i,8)');
simIn = simIn.setBlockParameter('Sim/TrainingsQuantityListener','Value', 'in(i,9)');
simIn = simIn.setBlockParameter('Sim/TrainingsQuantitySpeaker','Value', 'in(i,10)');

```

```
simIn = simIn.setBlockParameter('Sim/Proactivity','Value', 'in(i,11)');
```

```
pre_out=sim(simIn)
```

```
out=pre_out.simout.Data(1,1)
```

ПРИЛОЖЕНИЕ 6

Выдержка кода VBA из базы данных Access:

Option Compare Database

```
Private Sub Form_Load()
```

```
Me.RecordSource = "select * from Target where IDpurpose = " & Me.OpenArgs
```

```
Me.PurposeName = Me.OpenArgs
```

```
If Me.respsupp = 0 Then
```

```
    Me.respsupp = ""
```

```
End If
```

```
End Sub
```

```
Private Sub Back_Click()
```

```
DoCmd.OpenForm "Purpose"
```

```
DoCmd.Close acForm, "Target"
```

```
End Sub
```

```
Private Sub DeleteTarget_Click()
```

```
'Delete target from all tables
```

```
Me.Dirty = False
```

```
CurrentDb.Execute "Delete * from TargetResponsible where TargetResponsible.IDTresponsible in  
(select IDstuffmember from Target INNER JOIN TargetResponsible ON Target.IDtarget =  
TargetResponsible.IDtarget where Target.IDtarget = " & Me.IDtarget & ")"
```

```
CurrentDb.Execute "Delete * from TargetSupporter where TargetSupporter.IDTsupporter in (select  
IDstuffmember from Target INNER JOIN TargetSupporter ON Target.IDtarget =  
TargetSupporter.IDtarget where Target.IDtarget = " & Me.IDtarget & ")"
```

```
CurrentDb.Execute "Delete * from TargetYears where IDtarget = " & Me.IDtarget
```

```
CurrentDb.Execute "Delete * from Target where IDtarget = " & Me.IDtarget  
Me.Requery  
End Sub
```

```
Private Sub Target_AfterUpdate()  
IDpurpose.Value = Me.OpenArgs  
End Sub
```

```
Private Sub ToCIResponsible_Click()  
DoCmd.OpenForm "TargetResponsible", , , , , Me.IDtarget  
End Sub
```

```
Private Sub ToCISupporter_Click()  
DoCmd.OpenForm "TargetSupporter", , , , , Me.IDtarget  
End Sub
```

```
Private Sub ToStrategyCI_Click()  
DoCmd.OpenForm "StrategyCI", , , , , Me.IDtarget  
End Sub
```